WebSphere software

IBM

WebSphere Application Server V5 for iSeries:

Installation, Configuration, and Administration

- Covers both Base and Network Deployment editions
- All-in-one book for installation and configuration of WAS V5 on iSeries
- Lots of tips to make your work more efficient

Aleksandr Nartovich
Ursula Althoff
Greg Bobak
Diana Maribel Plazas
Arthur Pong
Mark Pottorff
Ted E. Pshock
Michael R. Spirito
David A. Thompson

Redbooks

ibm.com/redbooks

International Technical Support Organization

WebSphere Application Server V5 for iSeries: Installation, Configuration, and Administration

June 2003

SG24-6588-00

Note: Before using this information and the product it supports, read the information in “Notices” on page xi.

First Edition (June 2003)

This edition applies to Version 5.0 of WebSphere Application Server for iSeries.

Contents

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Trademarks

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

AIX®
AS/400®
AS/400e™
CICS®
DB2 Universal Database™
DB2®
Domino™
@server™
IBM.COM™
ibm.com®
IBM®
iSeries™
Lotus®
Lotus Notes®
MQSeries®
Netfinity®
Notes®
OS/400®
PartnerWorld®
pSeries™
Redbooks(logo)™ 
Redbooks™
RMF™
SOM®
SP™
Tivoli®
VisualAge®
WebSphere®
Word Pro®
xSeries™

The following terms are trademarks of other companies:

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.

Preface

IBM® WebSphere® Application Server for IBM @server™ iSeries™ (WAS) is an e-business application deployment environment built on open standards-based technology. It is the cornerstone of WebSphere offerings and services. In order to efficiently use WAS on iSeries customers need to master several skills:

- Installing and configuring the iSeries system for WAS
- Maintaining WAS on iSeries in the most efficient way
- Developing WebSphere applications according to Java 2 Platform, Enterprise Edition (J2EE) specification

This IBM Redbook will help you to gain the proficiency in installing, configuring and administering the WAS environment on iSeries. In order to learn the J2EE programming skills, refer to “Related publications” on page 553.

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Rochester Center.

Aleksandr V. Nartovich is a Senior I/T Specialist in the IBM International Technical Support Organization (ITSO) Rochester Center. He joined the ITSO in January 2001 after working as a developer in the IBM WebSphere Business Components (WSBC) organization. During the first part of his career, Aleksandr was a developer in AS/400® communications. Later, he shifted his focus to business components development on WebSphere. Aleksandr holds two degrees: in Computer Science from the University of Missouri-Kansas City and in Electrical Engineering from Minsk Radio Engineering Institute. You can reach Aleksandr at: alekn@us.ibm.com®

Ursula Althoff is an iSeries System Engineer working at IBM Enterprise System Sales Technical Support iSeries EMEA in Germany. She has worked at IBM for 26 years. Her experience on midrange computers started with S/38 and of cause AS/400 since it exists. Her areas of experience include OS/400®, application development, WebSphere Application Server on iSeries, and WebSphere Commerce for iSeries. She developed some courses about e-business on iSeries for IBM learning services (ILS) and wrote articles and some chapters in IBM Redbooks™ about this topics.

Greg Bobak has been a Programmer/Analyst for Cole National for 3 years. He holds a BS in Computer Science from Bowling Green State University. Greg is a WebSphere Commerce Suite 5.1 Implementation certified specialist.

Arthur Pong is an Advisory I/T Specialist in IBM Hong Kong and he has worked for IBM HK for 7 years. Before joining IBM, Arthur had five years experience on application development and project management experience. Arthur works in Technical Support Center now and his areas of expertise include iSeries, MQ, image products, WebSphere, programming, and teaching. Arthur holds master degrees in computer science and MBA.

David A Thompson is an IT Specialist with IBM Australia. He has 14 years of experience in the S/38 AS/400 and iSeries arenas. Joining IBM in New Zealand six years ago as a generalist on AS/400 supporting all aspects of the AS/400 system from installation to remote support and customization. Transferring to Australia in January 2001 David now works as part of the A/NZ support team covering specifically WebSphere and generally communications related problems. He can be reached at dathomps@au1.ibm.com.

Diana Maribel Plazas is an IBM Certified IT Specialist on iSeries, in Bogotá,Colombia. She has 8 years of experience on iSeries platform. She works on ITS unit doing post sale support and services on iSeries area. She is working also with the e-business group to improve the use of WebSphere in Colombia. She has experience with WebSphere on different customers in Colombia, and also on other areas from iSeries. She hold post graduated studies on Project manager and she has worked as a technical leader on many customers in her country.

Ted E. Pshock is a System Specialist for Cole National Corporation of Twinsburg Ohio. Ted has worked in the computer industry for 18 years and was an IBM AS/400 consultant for Affiliated Resource Group of Columbus Ohio. His current area of responsibilities includes IBM original HTTP server, WebSphere Commerce Suite, WebSphere Application Server, MQ Series and backend integration using RPGLE on the IBM iSeries Server. In addition, Ted is IBM certified AS/400 developer.

Mark Pottorff is a Consulting Specialist in the IBM Advanced Technical Support organization in Rochester, MN. He has provided technical support to the Business Partner and ISV members of PartnerWorld® for Developers for the past 9 years. His areas of expertise include: WebSphere Application Server, iSeries Application Development, Database, Performance, and Client/Server computing. Prior to that, he was head of software development for an IBM Business Partner. Mark has also presented at the Common User's Group. He holds a Bachelors degree in Computer Science.

Michael R. Spirito is an e-Business Systems Analyst at Cole National's Twinsburg, Ohio location. He earned a degree in Computer Science from Tri State in Erie, Pennsylvania. He has over five years of experience utilizing backend integration and Web technology. His areas of expertise include the iSeries Server, WebSphere Application Server, WebSphere Commerce Suite, WebSphere MQ, and RPGLE. For the past three years he has worked extensively with Application Server and Commerce Suite.

Special thanks to **Frances Stewart**, IBM Rochester, in the development lab, for her help in answering endless questions of the ITSO team.

Thanks to the following people for their contributions to this project:

Carla Sadtler
International Technical Support Organization, Rochester Center

Yvonne Lyon
International Technical Support Organization, San Jose Center

Art Smet
Brian Fewell
Byron L Bailey
Daniel Hiebert
Jim Fall
Kevin Paterson
Larry Hall
Les Fullem
Lisa Wellman

Lynn Boger
Melissa Anderson
Michael C Turk
Michael R Burke
Pat Fleming
Roshmi Bhaumik
Sharad Cocasse
Steve Simonson
IBM Rochester, Minnesota

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our Redbooks to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- Use the online **Contact us** review redbook form found at:

  ibm.com/redbooks

- Send your comments in an Internet note to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. JLU Building 107-2
  3605 Highway 52N
  Rochester, Minnesota 55901-7829

1

Introduction to WebSphere Application Server V5.0 for iSeries

IBM WebSphere Application Server, Version 5.0, delivers flexible configuration and deployment options to meet needs for stand-alone or multiserver distributed and highly dynamic environments. As your e-business requirements change, you can migrate smoothly to the greater functionality and higher qualities of service offered by other configurations.

IBM WebSphere Application Servers have several major themes that drive its new functionality and services:

- J2EE 1.3 compliance
- Web services
- JMX administration model

In this chapter we provide an overview of the WebSphere Application Server V5.0 implementation on the iSeries Server. The iSeries implementation has some distinct differences from the other platforms.

1.1 New packaging for WebSphere editions

Figure 1-1 shows the packaging modules of WebSphere Application Server V5.0 for iSeries:

- Express
- Base
- Network Deployment

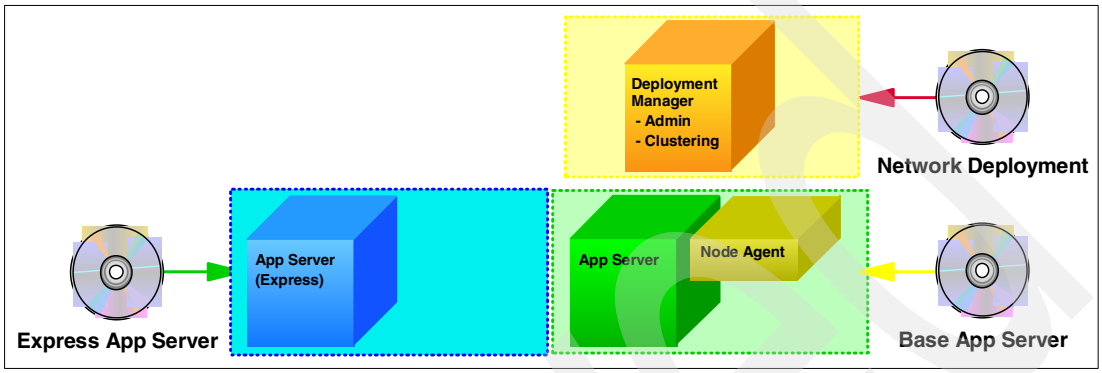


Figure 1-1 WebSphere Application Server 5.0: packaging modules

Several packaging scenarios are offered for these modules.

1.1.1 WebSphere 5.0 packaging scenarios

There are three packaging scenarios within WebSphere Application Server V5.0 for iSeries (see Figure 1-2):

1. **IBM WebSphere Application Server — Express for iSeries:** This includes the Express application server module.
2. **IBM WebSphere Application Server V5.0 for iSeries (Base edition):** This includes the Base application server module.
3. **IBM WebSphere Application Server Network Deployment V5.0 for iSeries:** This includes the Base application server and Network Deployment (ND) modules.

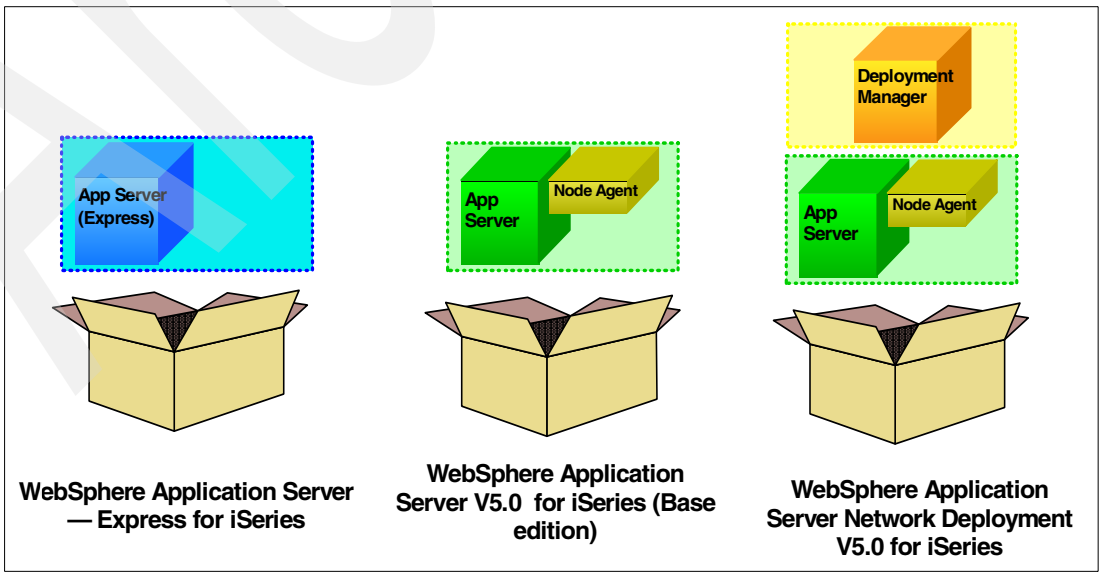


Figure 1-2 WebSphere packaging

1.2 New features

There are a number of significant differences between WebSphere Application Server V5.0 for iSeries and the previous versions of WebSphere Application Server for iSeries. In the following sections we outline the most important of these differences.

1.2.1 WebSphere Application Server V5.0 for iSeries (Base edition)

The base product provides an application server, which includes the runtime, administration interfaces, and application programming interfaces (APIs) for a single server installation (or single split-server installation). WebSphere Application Server Version 5.0 includes these features:

- J2EE 1.3 compliance:

  J2EE compliance has been one of IBM's main motivations for building WebSphere V5.0. It was necessary for IBM to comply with the latest J2EE standards. IBM wants to support the latest trends and industry standards as they become available, and the new specifications have significant improvements that IBM can use to increase the capabilities of the WebSphere product.

  The J2EE 1.3 specification has provided new development specifications. Servlet 2.3, JSP 1.2, and EJB 2.0 provide added API libraries for distributed application processing as well as integrated messaging support. More details on these specifications can be found in the J2EE Technologies Section.

- Java Management Extensions (JMX) administration model:

  IBM has chosen to redesign the administrative model in WebSphere to adopt a standard resource management interface. JMX is a new framework that has been added to the Java language. JMX allows you to wrap all of your resources (hardware and software) in Java and expose them in a distributed environment. JMX also provides a mapping framework for integrating existing management protocols, such as SNMP, into JMX's own management structures.

  Some of the benefits of using a JMX administrative model are:

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

- All configuration data is stored in XML files.
- An integrated Java Message Service (JMS) server. JMS is part of the J2EE 1.3 specification, and enables asynchronous messaging between application components.
- Web Services:

  Another motivation for the basis of WebSphere is the improvement of IBM's provision for Web Services. WebSphere has improved its SOAP support in this release and also ships with the Private UDDI. The private UDDI allows customers to become the service provider without having to comply with any additional requirements by the platform. They can use the Private UDDI to publish their services directly, rather than have a third-party broker provide that service. IBM has also included a Web Services Gateway which will allow customers to integrate with other Web Services in heterogeneous Web Services environments. More details on the enhancements for Web Services can be found under the "What's New in WebSphere" section.

- Dynamic network caching.
- HTTP session state failover support.

- A session manager is integrated into each application server.
- New security features are provided, including support for JAAS, CSIv2 interoperability, Java 2 security, and third-party security providers. WebSphere v5.0 extends its security support by enhancing its authentication options to include Kerberos tokens. These tokens provide strong authentication security for client/server applications. If you would prefer to use an alternative security authentication or authorization security implementation, WebSphere 5.0 provides Security Programming Interfaces (SPIs) for integration into those third party solutions. There is the SPI policy file associated with the server. It is located at:

  ```
  <WAS instance root directory>/config/cells/<cellName>/nodes/<node name>/spi.policy
  ```

  Note: In previous versions of WebSphere, there was a sas.server.props properties file that could be used to configure security options provided by the Security Associations Services in WebSphere. For compatibility with previous versions, these files continue to exist. This properties file has been replaced by security.xml at the server and cell levels. The SAS service has been replaced by CSIv2.

  In addition, there are three files, at the cell level, that we can use to configure security:

  – security.xml
  – admin-authz.xml
  – naming-authz.xml

  Review Chapter 8, “Security” on page 347 for further security information.
- The Performance Monitoring Infrastructure (PMI) is integrated with JMX.
- Resource Analyzer has been rebranded as Tivoli® Performance Viewer and includes extended functionality.
- Troubleshooting includes First Failure Data Capture (FFDC), which allows the collection of data based on the first failure in the system. The data is collected automatically by the WebSphere Application Server runtime, and stored in log files for the WebSphere Application Server instance.
- New collector tool gathers system information and packages it in a JAR file that you can send to IBM Service for analysis.

1.2.2 WebSphere Application Server Network Deployment V5.0 for iSeries

Additional function is available through the Network Deployment option. The Network Deployment option extends the base product with support for distributed systems administration, clustering, and workload management, and advanced Web services.

- Clustering allows you to create a group of servers that work together to process client requests. Clustering supports these functions:
  – Workload management
  – Failover
  – Distributed security
  – Distributed naming
- With distributed systems administration, you can use a single administrative interface to manage multiple application servers in a clustered environment. The single administrative interface supports these functions:
  – Presentation of a single-system image
  – Distribution of configuration information
  – Distribution of applications throughout a Network Deployment cell
  – Monitoring for distributed systems

- WebSphere Application Server Network Deployment supports these advanced Web services features:
  - A private Universal Description, Discovery, and Integration (UDDI) registry
  - Web Services Gateway (WSGW) support
- The Edge components installation image (*installed on a separate workstation*) 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.

1.2.3 What's unique to WebSphere Application Server V5.0 for iSeries

The iSeries version of WebSphere Application Server has several unique features as compared to the distributed platform version:

- WebSphere Application Server V5.0 for iSeries (WAS) runs in its own subsystem. The subsystem name will vary, depending on the version and edition of WebSphere Application Server you work with. For WebSphere Application Server 5.0 for iSeries, the subsystem is QEJBAS5. The WebSphere Application Server Network Deployment V5.0 for iSeries runs in subsystem QEJBASND5.
- HTTP server is supported on iSeries server:

  In order the run servlets or JSPs under WebSphere Application Server V5.0, you need to configure an HTTP server instance. The following HTTP servers can be used on the iSeries server as a Web server with WebSphere Application Server V5.0:
  - IBM HTTP Server for iSeries (Powered by Apache)
  - The Lotus® Domino™ for AS/400 Version 5.0.5 or Version 6.0 HTTP Server
- JDK support:

  WebSphere Application Server V5.0 supports JDK 1.3.1. This is provided by the license program 5722JV1 option 5 Java Developer Kit 1.3.
- WebSphere Application Server V5.0 for iSeries and WebSphere Application Server Network Deployment V5.0 for iSeries provide the QShell scripts that allow you to easily change port numbers and display them.

1.3 What's not available from the previous release (v4)

The following items have been removed or changed between release V4 and release V5:

- The Administrative console is now Web based and can be viewed using a Web browser.
- WebSphere will also provide command line tools, as it has done with previous versions, but this time those tools have been standardized. The tool known as wscp has been replaced by a JMX alternative, *wsadmin*.
- The XMLConfig Tool has been removed, since the Admin database repository has been removed. There is no longer any need to have a tool that converts relational admin data to XML, now that the administrative data is stored in XML documents.
- The Original HTTP server is no longer supported, only the HTTP server (powered by Apache) and Domino HTTP server are supported in WAS v5.
- The sas.server.props property file has been replaced by security.xml at the server and cell levels. The SAS service has been replaced by CSIv2.
- Resource Analyzer is replaced by Tivoli Performance Viewer.
- XML4J2.0.15 APIs are no longer supported.

Restrictions:

- V5.0 systems management does not manage V3.5 or V4 nodes.
- V5.0 clients cannot be Workload Manager EJB workload balanced to V3.5.x Servers.

1.4 Overview of the architecture

To complete our introduction to WebSphere Application Server V5.0 for iSeries, we discuss the architecture of the new version of WebSphere Application Server. This section provides the high level picture of the WAS architecture. It will greatly help you in understanding the topics described in this book.

1.4.1 WebSphere Application Server V5.0 for iSeries (Base edition)

Figure 1-3 shows the high-level architecture of WebSphere Application Server V5.0 for iSeries.

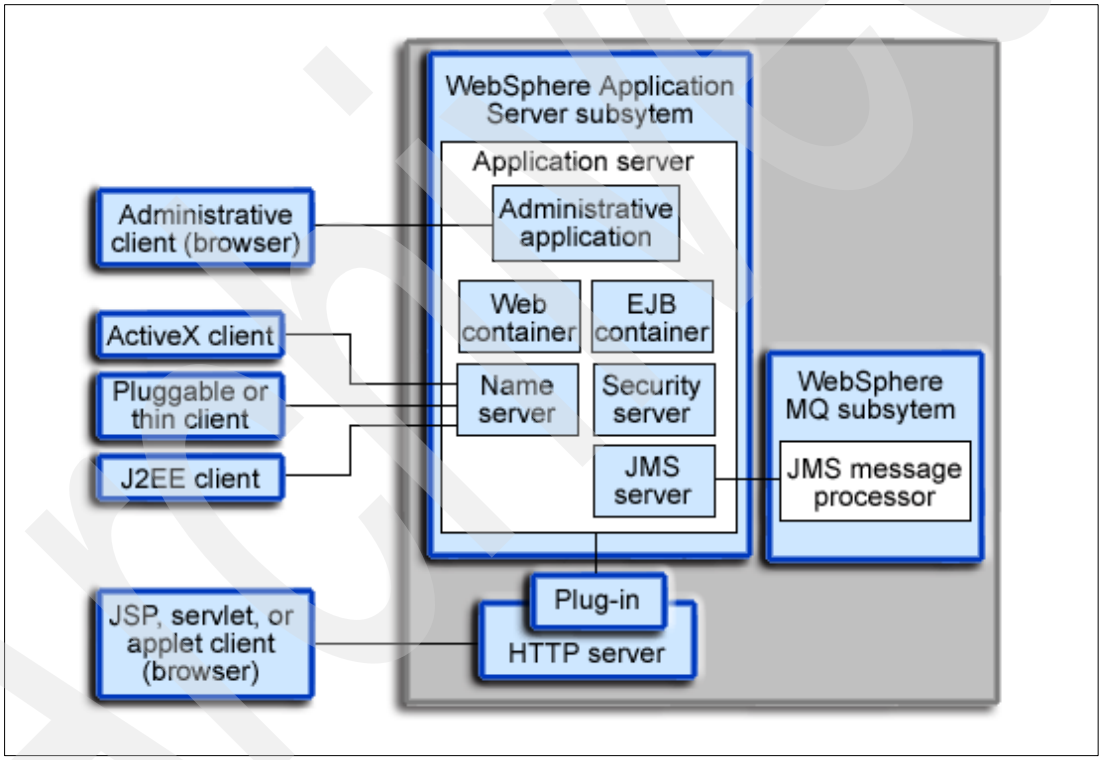


Figure 1-3 The WebSphere Application Server V5.0 for iSeries architecture

The WebSphere Application Server architecture consists of these software components that run on iSeries:

- WebSphere Subsystem:

  The WebSphere Application Server subsystem, QEJBAS5, contains the jobs that pertain to WebSphere Application Server. In addition to the jobs running in the QEJBAS5, your applications may also use jobs running in other subsystems. For more information on the jobs used in WebSphere Application Server, see WebSphere Application Server jobs.

- Application Server:
  - Web Container:

    The Web container runs within the application server and handles requests for servlets, JavaServer Pages (JSP), and the Web applications that contain them.
  - EJB Container:

    The application server interacts with the EJB container to allow access to the enterprise beans contained within the EJB container. The EJB container provides an interface between the enterprise beans and the application server, providing many low-level services such as threading, support for transactions, and management of data storage and retrieval.
  - Java Message Service (JMS) Server:

    WebSphere Application Server supports asynchronous messaging based on the Java Message Service (JMS) specification version 1.0.2 and supports the Application Server Facility (ASF) function defined within that specification. WebSphere Application Server provides an internal JMS provider and administration objects for MQSeries® as the JMS provider. You can use the internal JMS provider, install the MQSeries JMS on top of the WebSphere internal JMS, or install and configure another JMS provider.

    For an *unfederated* (also known as standalone) node, the internal JMS server runs within the application server. An unfederated node is an instance that is not part of a Network Deployment domain (cell). A *federated* node is an instance that has been added to a Network Deployment domain using the Network Deployment administrative console or the addNode script.
  - Name Server:

    The Java Naming and Directory Interface, or JNDI, is used to provide access to Java components within a distributed computing environment. The WebSphere Application Server name server provides the implementation of this J2EE service, allowing you to bind WebSphere Application Server resources to JNDI names, and allowing client applications to access resources such as data sources, enterprise beans, message listeners, etc.
  - Security Server:

    The WebSphere Application Server security server provides security infrastructure and mechanisms to protect sensitive J2EE resources and administrative resources and to address enterprise end-to-end security requirements on authentication; on resource access control; on data integrity, confidentiality, and privacy; and on secure interoperability.
- MQ Series subsystem:

  If you are using the embedded JMS server which comes with the WebSphere Application Server product, the MQ Series subsystem, QMQM, contains jobs that pertain to WebSphere Application Server. These jobs represent the embedded JMS server and are used to process JMS messages. For more information on the jobs running in the QMQM subsystem to support the embedded JMS server, see Other jobs used by WebSphere Application Server.

These are the software components that run on a client's workstation:

- Browser:

  A Web browser that supports HTML 4.0 and CSS Cascading Style Sheets (CSS) and has cookies enabled.

- Administrative Console:

  The Administrative Console is a browser-based graphical interface that allows you to configure and manage WebSphere Application Server resources. For more information about the Administrative Console, see A.5, “Administrative Console” on page 521.

1.4.2 WebSphere Application Server Network Deployment V5.0 for iSeries

Figure 1-4 shows the high-level architecture of WebSphere Application Server Network Deployment V5.0 for iSeries.

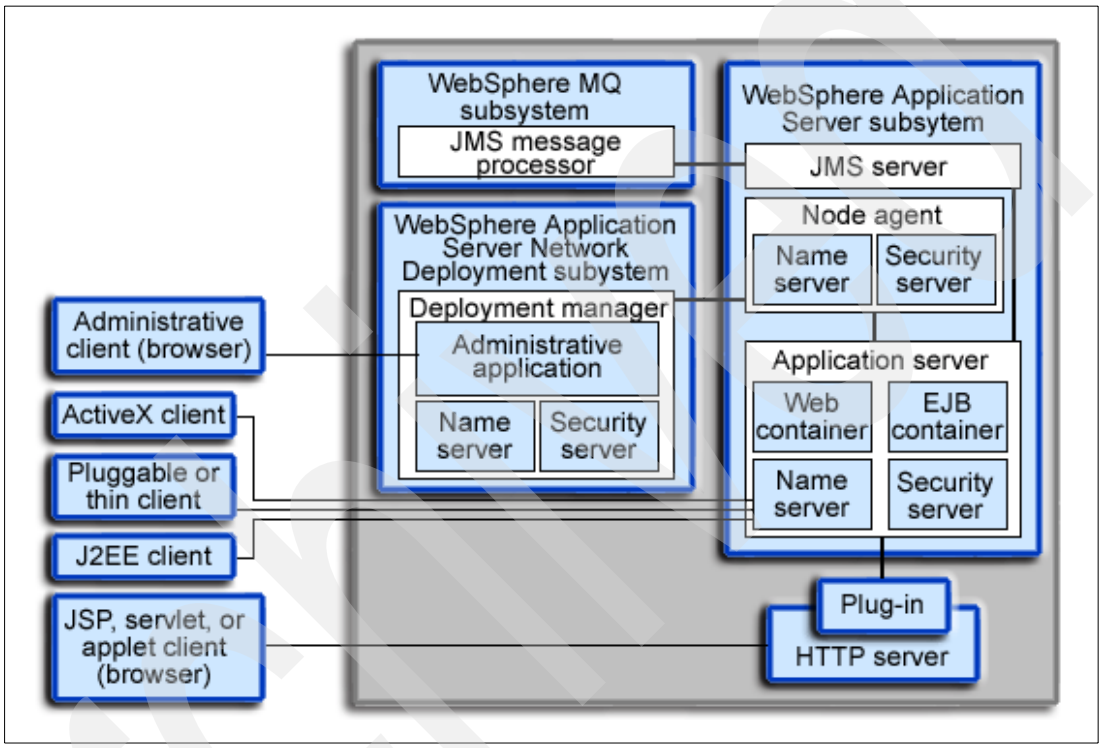


Figure 1-4 The WebSphere Application Server Network Deployment V5.0 for iSeries architecture

In addition to the components that make up the WebSphere Application Server V5.0 for iSeries architecture, the WebSphere Application Server Network Deployment V5.0 for iSeries architecture includes these software components for iSeries:

- Network Deployment Subsystem:

  The WebSphere Application Server Network Deployment subsystem, QEJBASND5, contains the jobs that pertain to Network Deployment. The QEJBASND5 subsystem contains the application server job for each Deployment Manager running on your iSeries.

- Deployment Manager:

  The Deployment Manager is an administrative application that runs in a specialized application server. The application server and the administrative application are installed when you install the WebSphere Application Server Network Deployment product. The Deployment Manager allows you to manage multiple WebSphere Application Server instances, or nodes, as a group. This logical grouping of nodes, managed by a single Deployment Manager instance, form a Network Deployment cell.

  Deployment managers also host logic for creating and controlling clusters, and for balancing the work load of application servers across several nodes.

  Administrative tools that need to access any managed resource in a domain usually connect to the deployment manager for the domain as a central point of control. The deployment manager provides a single, central point of administrative control for all elements of the entire WebSphere distributed cell.

- Node Agent:

  Node agents are servers that monitor application servers running in a WebSphere Application Server instance and route administrative requests to the application servers. If your WebSphere Application Server instance is not part of a Network Deployment cell, there is no node agent server for the instance.

  There is one node agent for each WebSphere Application Server instance that has been added to a Network Deployment cell.

  The node agent is used purely for administrative functions and is not involved in application serving functions. A node agent also hosts other important administrative functions such as file transfer services, configuration synchronization, and performance monitoring.

- JMS Server:

  WebSphere Application Server supports asynchronous messaging based on the Java Message Service (JMS) of a JMS provider that conforms to the JMS specification version 1.0.2 and supports the Application Server Facility (ASF) function defined within that specification. WebSphere Application Server provides an internal JMS provider and administration objects for MQSeries as the JMS provider. You can use the internal JMS provider, install the MQSeries JMS on top of the WebSphere internal JMS, or install and configure another JMS provider.

  For a federated node, the internal JMS server runs in its own job separate from application server process. A federated node is an instance that is part of a Network Deployment cell.

1.5 Development environment for WAS V5.0

There is a new excellent development tool: WebSphere Development Studio Client Advanced Edition for iSeries, Version 5.0. It leverages the capabilities of WebSphere Studio Application Developer to optimize and simplify J2EE application development. This product features best practices, templates, code generation, and one of the most comprehensive development environments in its class. It includes the iSeries suite of tools for traditional, client/server and e-business development. Figure 1-5 shows the major components of WebSphere Development Studio Client Advanced Edition for iSeries, Version 5.0.

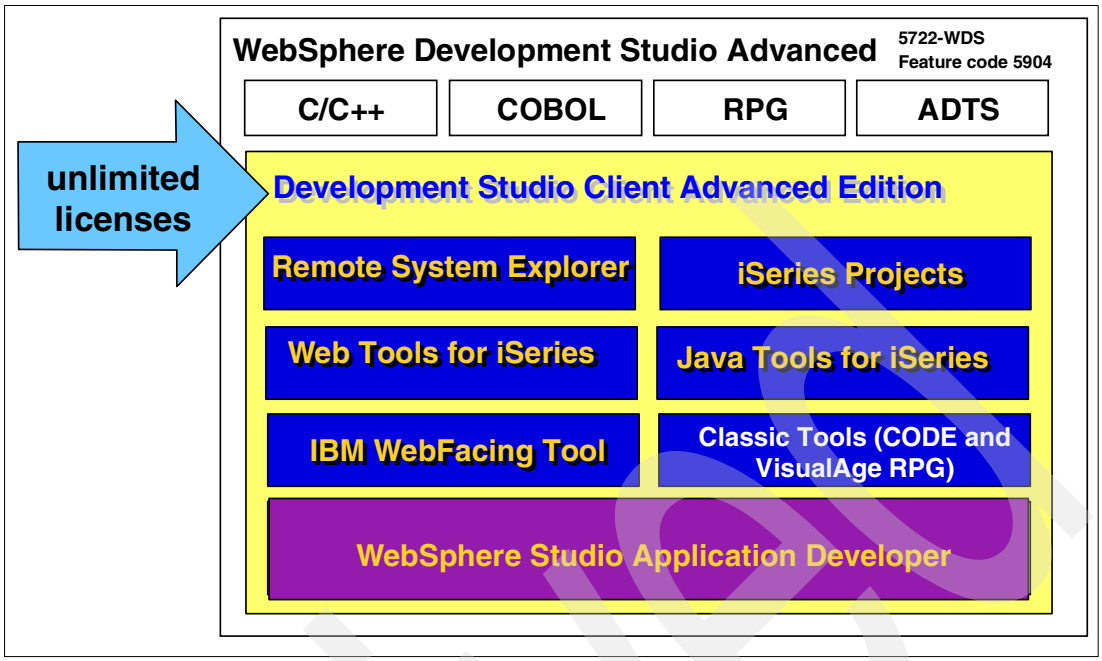


Figure 1-5 The tool's components

For more information about this tool, see the following Web page:

http://www.ibm.com/software/awdtools/wds400/

2

Installation of WebSphere Application Server 5.0 for iSeries

In this chapter we describe how to install WebSphere Application Server V5.0 (WAS) on an iSeries server. WAS requires a minimum hardware configuration. A number of prerequisite software licensed program products are also required.

Since WAS can be installed in a number of different ways, we provide a summary of the decisions that must be made when planning the installation. This chapter is organized into the following sections:

- **Prerequisites for WAS on iSeries:** See 2.2, “Prerequisites for WebSphere Application Server on iSeries” on page 13.
- **Planning your installation:** See 2.3, “Planning installation for WAS for iSeries” on page 15.
- **Performing WebSphere installation:** See 2.10, “Installing WebSphere Application Server 5.0 for iSeries” on page 25.
- **Performing Post installation tasks:** See 2.15, “Verifying TCP/IP configuration” on page 43
- **Uninstalling WebSphere:** See 2.23, “Deleting WebSphere Application Server components” on page 55.

2.1 What products and options are included?

In order to make an informed decision about what options of WebSphere Application Server you need to install, read this section. It provides an overview of the products and installation options that are part of the WebSphere Application Server V5 for iSeries. By default, all products and all options are installed. You can install a subset of products and options.

The installation CDs include the following products:

- WebSphere Application Server V5.0 for iSeries (5733WS5)
- WebSphere MQ V5.3 (5724B41)
- WebSphere MQ classes for Java and JMS V5.3 (5639C34)

WebSphere Application Server v5.0 for iSeries (5733WS5) options

Decide what options you need for WAS:

- *Option Base*: WebSphere Application Server:

  This option is required. It contains the Readme file and other common files for the product.

- *Option 1*: Client development and run time:

  Option 1 provides the client application development and run time portion of the WebSphere Application Server V5.0 product. This option allows you to compile your client application code and run both Java 2 Enterprise Edition (J2EE) client applications and thin applications. It also provides the necessary scripts and runtime for running remote HTTP Web servers to access WebSphere Application Server, which is used in multiple-machine topologies.

- *Option 2*: Application server run time:

  For Option 2 to function correctly, option 1 must also be installed. Option 2 provides the application server run time for WebSphere Application Server. It allows you to deploy and run J2EE 1.3 compliant enterprise applications.

- *Option 3*: Samples:

  For Option 3 to function correctly, option 2 must also be installed. Option 3 provides the samples for WebSphere Application Server.

- *Language option:*

  The language option denotes the National Language Support (NLS) option for the product. Most of the common SBCS and DBCS are available.

WebSphere MQ V5.3 for iSeries (5724B41) options

Decide what options you need for WebSphere MQ:

- *Option Base*: WebSphere MQ:

  Install this product if you want to use the embedded JMS server component of WebSphere Application Server V5.0.

- *Option 1*: WebSphere MQ - Samples:

  Option 1 provides the samples for WebSphere MQ.

- *Language option:*

  The language option denotes the (NLS) option for the product. It is installed only for Option *BASE.

WebSphere MQ classes for Java and JMS V5.3 (5639C34) options

Decide what options you need for WebSphere MQ classes for Java and JMS:

- *Option Base*: WebSphere MQ classes for Java and JMS:

  Install this product if you want to use JMS from your applications or the embedded JMS server.

- *Option 1*: WebSphere MQ classes for Java and JMS - Samples:

  Option 1 provides the samples for WebSphere MQ classes for Java and JMS.

- *Language option:*

  The language option denotes the (NLS) option for the product. It is installed only for Option *BASE. Only 2924 - English is available.

2.2 Prerequisites for WebSphere Application Server on iSeries

Before you install WebSphere Application Server, verify that your hardware and software meet the minimum requirements.

2.2.1 iSeries and AS/400 hardware requirements

In order to size your iSeries server, to get help with estimating the system configuration, you can make use of the IBM Workload Estimator for iSeries:

(http://www-912.ibm.com/servlet/EstimatorServlet)

Systems not meeting the recommended minimums can be used in environments that support a limited number of users and where longer server initialization times are acceptable. The recommended HW is based on the type of the applications you're going to deploy to WAS:

- If your applications consist only of servlets and JavaServer Pages (JSP) files, your server should meet these requirements:
  - Recommended minimum server models:
    - AS/400e™ server 170 with processor feature 2292
    - AS/400e server 720 with processor feature 2061
    - iSeries Model 270 with processor feature 2250
    - iSeries Model 820 with processor feature 2395
  - 300 CPW per application server
  - 750 MB of memory (recommended minimum) per application server

- If your applications contain enterprise beans, your server should meet these requirements:
  - Recommended minimum server models:
    - AS/400e server 170 with processor feature 2385
    - AS/400e server 720 with processor feature 2062
    - iSeries Model 270 with processor feature 2252
    - iSeries Model 820 with processor feature 2396
  - 460 CPW per application server
  - 1 GB of memory (recommended minimum) per application server

Note:

1. These requirements are based on a single WebSphere Application Server instance. Additional instances running concurrently will require additional resources.
2. These requirements represent the recommended minimum requirements. Deployments which must support many users or require shorter response times may require additional resources.

You can find the disk requirements for the various products in Table 2-1.

Table 2-1 Disk requirements

WebSphere Application Server for iSeries		
Installation option	**Description**	**Disk space after installation**
*BASE	WebSphere Application Server	15 MB
Option 1	Client development and run time	400 MB
Option 2	Application server run time	200 MB
Option 3	Samples	25 MB
WebSphere MQ V5.3 for iSeries		
Installation option	**Description**	**Disk space after installation**
*BASE	WebSphere MQ	100 MB
Option 1	WebSphere MQ - Samples	2 MB
WebSphere MQ classes for Java and JMS V5.3 for iSeries		
Installation option	**Description**	**Disk space after installation**
*BASE	WebSphere MQ classes for Java and JMS	15 MB
Option 1	WebSphere MQ classes for Java and JMS - Samples	1 MB

2.2.2 iSeries and AS/400 software requirements

As a basic requirement, OS/400 Version 5 Release 1, or later (in an unrestricted state) is needed to install WebSphere Application Server 5.0. The following software is required:

Note: For V5Rx OS/400 license program, we use 5722-xxx as program product numbers.

iSeries and AS/400 required software

- OS/400 Version 5 Release 1 (V5R1) or Version 5 Release 2 (V5R2)
  The iSeries server must be in an unrestricted state, and your user profile must have sufficient authority. See 2.10.1, “Authority requirement” on page 25 for details.
- IBM Developer Kit for Java Version 1.3 (5722-JV1 option 5).
- OS/400 Qshell (5722-SS1 option 30)
  Required for local installation and to use scripts in WebSphere Application Server.

- OS/400 Host Servers (5722-SS1 option 12)
  Required for remote installation.
- HTTP server
  Not needed for installation, but required to support requests for servlets and JSP resources that you want to be served via WAS in an production environment. The HTTP server can be on the same iSeries server as WAS or on any other system. There are two HTTP server on iSeries available to work as an external HTTP server together with WAS:
  - IBM HTTP Server (powered by Apache) (5722-DG1)
  - Lotus Domino for AS/400 R5 (5769-LNT)

iSeries and AS/400 optional software

- OS/400 Digital Certificate Manager (5722-SS1 option 34)
  Not required for installation, but required if you plan to use Secure Sockets Layer (SSL) protocol.
- A Cryptographic Access Provider (5722-AC3)
  Not needed for installation, but required if you plan to use SSL.
- DB2®(R) Query Manager and SQL Development Kit for iSeries (5722-ST1)
  Can be helpful in developing client applications.

2.3 Planning installation for WAS for iSeries

Since WAS can be installed in a number of different configurations, this section provides a step-by-step summary of the decisions that need to be made as you plan your installation. Prior to performing an installation, you should consider each of the following options:

- Are you migrating an existing installation?
- Should the GUI or a silent installation be used?
- Should a typical or a custom installation be used?

2.3.1 Planning your steps and time needed for installation

Good planning is half of the success. Before you install WebSphere Application Server for iSeries, you can use Table 2-2 to estimate the amount of time to install WAS.

Table 2-2 WebSphere Application Server 5.0 for iSeries installation planning

Task	Estimated time
Planning the installation.	
Verifying hardware and software prerequisites; see 2.1.1, “iSeries and AS/400 hardware requirements” on page 20 and 2.1.2, “iSeries and AS/400 software requirements” on page 21.	20 minutes
Obtaining the product and current fixes; see 2.4, “Obtaining WebSphere Application Server and current fixes” on page 16.	up to 2 weeks
Consider if you need to install Java Messaging Service (JMS); see page 18.	
Check for previous installed WAS versions; see page 19.	5 minutes
Reading Release Notes® and migration instructions; see 2.7, “Reading product Release Notes for important information” on page 20.	1-2 hours
Evaluate the available installation options and determine which options you require; see 2.1, “What products and options are included?” on page 12.	15 minutes

Installing all prerequisite software; see 2.2.2, “iSeries and AS/400 software requirements” on page 14 and install the OS/400 cumulative PTF package; see 2.8, “Installing the correct cumulative PTF package” on page 21	1-2 hours
Configure TCP/IP; see 2.9, “Starting, configuring, and verifying TCP/IP” on page 21.	30 minutes
Installing WebSphere Application Server for iSeries; see 2.10, “Installing WebSphere Application Server 5.0 for iSeries” on page 25. The time you need for installation depends on the method used. Installing the product from the CD-ROM drive of your iSeries server takes less time than installing it from a remote workstation.	45-120 minutes
Installing WebSphere Application Server Group PTF; see 2.14, “Installing WebSphere group PTF” on page 41.	Up to 2 hours
Most likely, you will need to IPL the iSeries system	
Installing optional workstation-based tools; see Chapter 3, “Installation of workstation tools” on page 59.	15 minutes
Verifying the installation; see 2.15, “Verifying TCP/IP configuration” on page 43.	10 minutes
Starting the WebSphere Application Server environment for the first time; see Chapter 6.2, “Setting up the WebSphere Application base environment” on page 109.	30-40 minutes

Notes:

1. Depending on your server, the installation may take more or less time.
2. Our testing is done on a 820 machine with V5R2 OS/400 running. The time needed for installation of WebSphere Application Server 5.0 for iSeries, from the iSeries CD-ROM, was only about 30 minutes.

2.4 Obtaining WebSphere Application Server and current fixes

You will need to update your iSeries system with the required level of fixes: cumulative PTFs and WAS group PTFs.

2.4.1 Cumulative PTFs

WebSphere Application Server for iSeries recommends a minimum OS/400 cumulative PTF package level. To determine the prerequisite cumulative PTF package level for the version of WebSphere Application Server you plan to install, see the PTF pages on:

http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/services/service.htm

From the PTF page, follow these steps:

1. In the WebSphere Application Server 5.0 row, click the link for your OS/400 release level.
2. Click the Cumulative PTF Package link to see the minimum cumulative PTF package that is required.

To determine if the correct OS/400 cumulative PTF package is installed on your server, perform these steps:

1. Sign on to your server.

2. Enter the Display PTF Status (DSPPTF) command on an OS/400 command line. The Display PTF Status screen is displayed. This screen lists the PTFs that have been applied to your server.

   Figure 2-1 shows an example of the Display PTF status screen:

```
                                Display PTF Status
                                                              System: RCHASO2B
Product ID  . . . . . . . . . . . . . :   5722999
IPL source  . . . . . . . . . . . . . :   ##MACH#B
Release of base option  . . . . . . . :   V5R2MO L0O

Type options, press Enter.
  5=Display PTF details   6=Print cover letter   8=Display cover letter

     PTF                                                        IPL
Opt  ID       Status                                          Action
     TL02036  Temporarily applied                             None
     TL01302  Superseded                                      None
     TL01254  Superseded                                      None
     TL01226  Superseded                                      None
     TL01163  Superseded                                      None
     TL01114  Superseded                                      None
     TL01086  Superseded                                      None
     RE01148  Permanently applied                             None
     RE01089  Permanently applied                             None
                                                                    More...
F3=Exit   F11=Display alternate view   F17=Position to   F12=Cancel
```

Figure 2-1 Check current iSeries PTF Cum level

In the example, the first PTF that is listed, TL02036 (with a status of Temporarily applied), correlates to the cumulative PTF that is installed on the server.

Note: The formal name for cum TL02036 is C2036520, in the format of CYJJJVRM (Y=year, J=Julian date, V=Version, R=Release, M=Modification). In our example, the cumulative PTF is built on the 036 days of year 2002.

You should order and install the prerequisite OS/400 cumulative PTF package before installing WAS. For instructions on ordering PTFs, see the IBM eServer iSeries Support Fixes:

http://www-912.ibm.com/supporthome.nsf/document/17403848

2.4.2 WebSphere group PTF

Program temporary fixes (PTF) for the WebSphere Application Server product are shipped as group PTFs for iSeries. The latest WebSphere Application Server group PTF must be loaded and applied prior to starting WebSphere Application Server for the first time.

This group PTF includes the latest WebSphere Application Server PTFs that bring the product up to the latest WebSphere Application Server for iSeries level. Group PTF also contains fixes for other software components:

- IBM DB2 Universal Database™
- IBM Developer Kit for Java
- IBM HTTP Server

- WebSphere MQ
- WebSphere MQ classes for Java and JMS

These PTFs are not in the cumulative PTF packages and have to be installed after installing the WAS software and before starting the WAS environment.

Installing the group PTF requires an IPL of your iSeries server, so plan accordingly. To determine which group PTF you must order and install, see:

http://www.ibm.com/eserver/iseries/software/websphere/wsappserver/services/service.htm

Note that the group PTF numbers differ by WebSphere Application Server product and OS/400 release level.

Table 2-3 shows the group PTF numbers in relation to the OS/400 and WAS editions.

Table 2-3 Group PTF numbers

	V5R1	**V5R2**
WebSphere Application Server V5.0 for iSeries	SF99243	SF99245
WebSphere Application Server Network Deployment V5.0 for iSeries	SF99244	SF99246

The Network Deployment group PTF includes the WebSphere Application Server V5 group PTF, so if you order the ND group, you do not need to order the WAS group PTF. Also, the group PTFs must be applied after installing the products and before attempting to start the servers for the first time.

2.5 Considering if you need Java Messaging Service (JMS)

Asynchronous messaging provides a method for communication based on the Java Message Service (JMS) programming interface. JMS supports the development of message-based applications in the Java programming language, allowing for the asynchronous exchange of data and events throughout an enterprise.

JMS is the standard Java API for applications to use to perform messaging. In J2EE 1.3, the Java Messaging Service becomes an integral part of Java 2 Platform, Enterprise Edition (J2EE). JMS support enables J2EE applications, as JMS clients, to exchange messages asynchronously with other JMS clients by using JMS destinations (queues or topics). A J2EE application can explicitly poll for messages on a destination, then retrieve messages for processing by business logic beans.

JMS provides a common mechanism for Java programs (clients and J2EE applications) to create, send, receive, and read asynchronous requests, as JMS messages. The specification was developed by Sun Microsystems with help from IBM, other messaging vendors, and other application server vendors. The standard defines a package of Java Interfaces, to be implemented by the messaging vendor, or “provider” to use the J2EE terminology.

You have two choices to use JMS support with WebSphere on your iSeries server:

- Use the full function JMS server from WebSphere MQ Series for iSeries V5.3.
- Use the embedded JMS server in WAS.

The full function JMS server is part of WebSphere MQ Series for iSeries V5.3 (5733-A38) licence software. The JMS server is fully integrated with the server's administration and runtime and it is compliant with J2EE 1.3 compliance tests.

WebSphere MQ Series for iSeries V5.3 JMS resources can also be administered and configured via the WebSphere administrative console. This is similar to the system management support for the embedded JMS server.

If you already use MQSeries for AS/400 V4.2 (5769MQ2) or MQSeries for AS/400 V5.2 or lower (5733A38) you have to migrate to WebSphere MQ V5.3 for iSeries in order to use JMS with WebSphere Application Server. Older MQSeries versions are incompatible for use with WebSphere Application Server V5.0 for iSeries. To migrate to WebSphere MQ V5.3 for iSeries, see the migration instructions in the WebSphere MQ for iSeries documentation at:

http://publibfp.boulder.ibm.com/epubs/html/amqwac02/amqwac02tfrm.htm

Although WAS 5.0 provides a fully-compliant embedded JMS provider, we expect that many customers will continue to use a full-blown WebSphere MQ as their JMS provider for either of two reasons:

- First of all, the embedded WAS JMS provider can only be used from within the WAS environment. Many customers are using WebSphere MQ V5.3 today for heterogeneous integration, allowing their J2EE applications to communicate with other MQ applications, that may be written in other languages (for example C, C++, Visual Basis and so on) and run on other platforms (Microsoft COM, for example).
- The other reason for using WebSphere MQ is in order to take advantage of the more advanced messaging topologies.

For more information on WebSphere MQSeries, see:

http://www.ibm.com/software/ts/mqseries/messaging/v53/

WebSphere Application Server provides an embedded JMS server that can be installed along with a WebSphere Application Server install. The embedded JMS Provider is fully compliant to the specifications. Since the built-in (embedded) JMS provider is a reduced footprint of MQSeries, there are limitations to using it; for example:

- The licence for WebSphere MQ V5.3 that comes with WebSphere Application Server V5.0 for iSeries doesn't allow a non-WAS use of the product.
- Primarily, it is not possible to communicate with non-WebSphere environments using the internal JMS provider.
- Some advanced options such as QMgr/QMgr channels, message flows and message transformations are not supported.
- The internal JMS Provider is not interoperable with WebSphere MQ, and therefore is not useful for heterogeneous communication between WebSphere Application Server and other environments.

2.6 Checking for previously installed versions of WAS and MQSeries

Before you start the installation of the products, check for the previously installed versions of WAS and MQSeries. Determine if WebSphere Application Server for iSeries is already installed on your server. Perform these steps:

1. Enter the Display Software Resources (DSPSFWRSC) command on an OS/400 command line.

a. Look for an entry with the product Resource IDs 5733WS5.

b. If you do not find the product Resources ID, then this product has not been installed on your iSeries server.

c. If a previous version of WebSphere Application Server is installed on your server, then, before you install version 5.0 of WebSphere Application Server, read the coexistence instructions at:

   http://www.ibm.com/eserver/iseries/software/websphere/wsappserver/product/coexist50.html

d. If you plan to migrate from the previous version to this version, be sure to read the migration instructions at:

   http://www.ibm.com/eserver/iseries/software/websphere/wsappserver/product/50migration.html

2. Determine if WebSphere MQ for iSeries is already installed on your server. Perform these steps:

   a. Enter the Display Software Resources (DSPSFWRSC) command on an OS/400 command line.

   b. Look for an entry with the product Resource ID 5724B41.

      If you do not find the product Resources ID, then this product has not been installed on your iSeries server.

   c. If MQSeries for AS/400 has been installed on your system, then you must migrate it to WebSphere MQ V5.3 for iSeries in order to use JMS with WebSphere Application Server. MQSeries for AS/400 V4.2 (5769MQ2) and MQSeries for AS/400 V5.2 (5733A38) are both incompatible for use with WebSphere Application Server. To migrate MQSeries for AS/400 to WebSphere MQ V5.3 for iSeries, see the migration instructions in the WebSphere MQ for iSeries documentation:

      (http://www.ibm.com/software/ts/mqseries/library/manualsa/)

      The WebSphere Application Server 5.0 for iSeries CD-ROM contains the WebSphere MQ V5.3 for iSeries product and can be used for the installation described in the migration documentation.

3. Determine if WebSphere MQ classes for Java and JMS V5.3 for iSeries is already installed on your server. Perform these steps:

   a. Enter the Display Software Resources (DSPSFWRSC) command on an OS/400 command line.

   b. Look for an entry with the product Resource ID 5639C34.

   c. If you do not find the product Resources ID, then this product has not been installed on your iSeries server.

   d. If MQSeries classes for Java and JMS (5648C60) has been installed on your system, then you must uninstall it before installing WebSphere MQ classes for Java and JMS (5639C34). To uninstall MQSeries classes for Java and JMS, enter this command:

      ```
      DLTLICPGM LICPGM(5648C60)
      ```

2.7 Reading product Release Notes for important information

Read the product Release Notes for important information about the product. For links to the Release Notes, see the WebSphere Application Server documentation page at:

http://www.ibm.com/eserver/iseries/software/websphere/wsappserver/docs/docws50.html

2.8 Installing the correct cumulative PTF package

WebSphere Application Server for iSeries recommends a minimum OS/400 cumulative PTF package level. If the correct cumulative PTF package is not installed on your iSeries server, you must install it before installing WebSphere Application Server.

Note: The cumulative PTF package requires a restart of your iSeries server. If it is not convenient to restart your server, you can simply load and apply the PTF specifying that the PTFs requiring an IPL be applied at the next normal IPL of the server. However, you should not install the WebSphere Application Server product until all of the PTFs have been successfully applied.

For information on verifying your cumulative PTF level, see 2.4.1, “Cumulative PTFs” on page 16.

To install the cumulative PTF, just follow the normal PTF installation procedures. For details, refer to:

http://publib.boulder.ibm.com/iseries/v5r2/ic2924/info/rzam8/rzam8fixinstallcum.htm

2.9 Starting, configuring, and verifying TCP/IP

To configure and run WebSphere Application Server on the iSeries, TCP/IP must be configured properly and must be started before you start the WebSphere Application Server environment. Here we describe only the basic configuration for TCP/IP on iSeries.

Note: For further information about TCP/IP Configuration and Installation, refer to the iSeries Information Center:

http://publib.boulder.ibm.com/iseries/v5r2/ic2924/index.htm

Then navigate to the Network>TCP/IP>TCP/IP Setup Section.

2.9.1 Basic configuration of TCP/IP on iSeries

To configure or verify TCP/IP on iSeries, use the Configure TCP/IP (CFGTCP) command. You can also use the iSeries Navigator, which provides a graphical interface to do this.

To configure or verify the TCP/P configuration follow, these steps:

1. Start the Configure TCP/IP menu by typing in CFGTCP on an OS/400 command line and pressing Enter.
2. Verify if your iSeries server has a TCP/IP host and domain name assigned:
   a. On the Configure TCP/IP menu, select option 12 (Change TCP/IP domain information).
   b. Verify that the TCP/IP host name and domain name is correct. If the host name or domain name is not correct, type the correct host name in the **Host name** field and the domain name in the **Domain name** field, and press Enter; for an example, see Figure 2-2. If all is correct, press F3 to return to the Configure TCP/IP menu.

```
                        Change TCP/IP Domain (CHGTCPDMN)

Type choices, press Enter.

Host name  . . . . . .   'RCHAS02B'

Domain name  . . . . . .   'IBM.COM™'



Domain search list . . . . . .   *DFT



Host name search priority  . . .   *LOCAL        *REMOTE, *LOCAL, *SAME
Domain name server:
  Internet address . . . . . .   '1.2.3.76'
                                 '1.2.3.75'



                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
F13=How to use this display        F24=More keys
```

Figure 2-2 Verify your TCP/IP host and domain name

Note: Changes of the host or domain name take only effect after an IPL (iSeries system startup.

3. Verify that your TCP/IP interfaces are defined and active:
   a. On the Configure TCP/IP menu, select option 1 (Work with TCP/IP interfaces).
   b. Here you define the internet address, subnet mask, and line description to use for the communication adapter (see Figure 2-3).

```
                        Work with TCP/IP Interfaces
                                                            System:   RCHAS02B
Type options, press Enter.
  1=Add   2=Change   4=Remove   5=Display   9=Start   10=End

     Internet        Subnet             Line      Line
Opt  Address         Mask            Description  Type

     1.2.3.4         255.255.255.128  ETHLINE      *ELAN
     127.0.0.1       255.0.0.0        *LOOPBACK    *NONE




                                                                      Bottom
F3=Exit      F5=Refresh   F6=Print list   F11=Display interface status
F12=Cancel   F17=Top      F18=Bottom
```

Figure 2-3 TCP/IP interface definition

c. Press F11 to display the interface status; see Figure 2-4.

Note: The TCP/IP interfaces can only be started when you have already started TCP/IP on iSeries; see 2.9.2, “Verifying that the server's IP address is associated with the host name” on page 25. There is also an autostart parameter in the interface definition. When this parameter is set to *YES, the interface is started when TCP/IP is started on iSeries.

d. Verify that the TCP/IP address is active. If it is not active, specify option 9 (Start).

e. Verify that the LOOPBACK interface with IP address 127.0.0.1 is active. If it is not active, specify option 9 (Start) next to the entry with IP address **127.0.0.1** and press Enter.

```
                        Work with TCP/IP Interfaces
                                                            System:   RCHAS02B
Type options, press Enter.
  1=Add   2=Change   4=Remove   5=Display   9=Start   10=End

     Internet         Subnet           Interface
Opt  Address          Mask             Status

     1.2.3.4          255.255.255.128  Active
     127.0.0.1        255.0.0.0        Active




                                                                       Bottom
F3=Exit      F5=Refresh   F6=Print list   F11=Display line information
F12=Cancel   F17=Top      F18=Bottom
```

Figure 2-4 Verify TCP/IP address and LOOPBACK interface are active

f. Press F3 to return to the Configure TCP/IP menu.

4. Define TCP/IP host table entries in the TCP/IP host table

   The host table on iSeries contains the IP addresses and associated names of the hosts in your TCP/IP network. A common practice is to define a long name (such as SYS1.ABC.COM) that conveys the network organization and also to define a short name (such as SYS1) that is easier to remember here.

   The host name of your iSeries server (as you defined in Figure 2-2 on page 22) has to be defined in this table. Update the host table entries as needed using the following instructions:

   a. On the Configure TCP/IP menu, select Option 10 (Work with TCP/IP host table entries).

   b. Use this display to add entries to the host table and to change, remove, rename, display, or print the entries in the table; see Figure 2-5 on page 24.

   c. When you change or edit an entry, type your IP address in the **Internet Address** field and type the host name in the **Name** field. You can enter more than one host name here, as in our example, where we define one entry for the host name and one entry for the host/domain name of our iSeries server.

```
                    Work with TCP/IP Host Table Entries
                                                              System:   RCHAS02B
Type options, press Enter.
  1=Add   2=Change   4=Remove   5=Display   7=Rename

     Internet         Host
Opt  Address          Name
     1.2.3.4          RCHAS02B
                      RCHAS02B.IBM.COM
     127.0.0.1        LOOPBACK
                      LOCALHOST



                                                                        Bottom
F3=Exit   F5=Refresh   F6=Print list   F12=Cancel   F17=Position to
```

Figure 2-5 TCP/IP Host Table Entries

To start TCP/IP manually, use the start TCP/IP (STRTCP) command on an OS/400 command line and press Enter.

You can automate the process to start TCP/IP at iSeries system startup by making use of the Change IPL Attributes (CHGIPLA) command (see Figure 2-6). Here, you set the Start TCP/IP parameter (STRTCP) to *YES.

```
                         Change IPL Attributes (CHGIPLA)

Type choices, press Enter.

Restart type . . . . . . . . . . RESTART        *SYS
Keylock position . . . . . . . . KEYLCKPOS      *NORMAL
Hardware diagnostics . . . . . . HDWDIAG        *MIN
Compress job tables  . . . . . . CPRJOBTBL      *NONE
Check job tables . . . . . . . . CHKJOBTBL      *ABNORMAL
Rebuild product directory  . . . RBDPRDDIR      *NONE
Mail Server Framework recovery   MSFRCY        *NONE
Display status . . . . . . . . . DSPSTS         *ALL
Start TCP/IP . . . . . . . . . . STRTCP         *YES
Clear job queues . . . . . . . . CLRJOBQ        *NO
Clear output queues  . . . . . . CLROUTQ        *NO
Clear incomplete joblogs . . . . CLRINCJOB      *NO
Start print writers  . . . . . . STRPRTWTR      *YES
Start to restricted state  . . . STRRSTD        *NO



                                                                        Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 2-6 Change IPL attributes

2.9.2 Verifying that the server's IP address is associated with the host name

Enter the ping command on an OS/400 command line together with the hostname or IP address. Here are two examples:

```
ping RCHASO2B
ping '1.2.3.4'
```

Verify that the ping is successful and the resulting IP address is correct.

Example 2-1 shows a sample output from a successful ping command.

Example 2-1 Sample output of the ping command

```
ping RCHASO2B
Verifying connection to host system RCHASO2B at address
    1.2.3.4.
  PING reply 1 from 1.2.3.4 took 0 ms. 256 bytes. TTL 64.
  PING reply 2 from 1.2.3.4 took 0 ms. 256 bytes. TTL 64.
  PING reply 3 from 1.2.3.4 took 0 ms. 256 bytes. TTL 64.
  PING reply 4 from 1.2.3.4 took 0 ms. 256 bytes. TTL 64.
  PING reply 5 from 1.2.3.4 took 0 ms. 256 bytes. TTL 64.
  Round-trip (in milliseconds) min/avg/max = 0/0/0
  Connection verification statistics: 5 of 5 successful (100 %).
```

2.10 Installing WebSphere Application Server 5.0 for iSeries

You can use one of these installation methods to install WebSphere Application Server on your iSeries server:

- From the CD-ROM drive of your iSeries server:

  Installing the product from the CD-ROM drive of your iSeries server requires direct physical access to the server. The local installation requires less time to complete than a remote installation. You start the local installation with either the Qshell SETUP script or the Run Java (RUNJVA) command. Refer to 2.11, "Installing WAS from CD-ROM drive of your iSeries server" on page 25 for detailed information.

- From the CD-ROM drive of a workstation with Microsoft Windows 32-bit operating system:

  Installing the product from the CD-ROM drive of a workstation does not require direct physical access to the iSeries server. The remote installation requires more time to complete than a local installation. Refer to Table 2-4, "Installation parameters" on page 29 for detailed information.

Note: The total installation may take up to 120 minutes to complete.

2.10.1 Authority requirement

To install WebSphere Application Server, you need an iSeries user profile where the user class needs to be set to *SECOFR and the special authorities to *USRCLS.

2.11 Installing WAS from CD-ROM drive of your iSeries server

WebSphere Application Server V5.0 for iSeries consists of two CDs, and the installer will be prompted to insert CD 2. You will be prompted to swap CDs.

You can perform an installation of WebSphere Application Server with the SETUP script in Qshell or with the Run Java (RUNJVA) command on an OS/400 command line.

To install WebSphere Application Server from the CD-ROM drive of your iSeries server, follow these steps:

- Sign on to the iSeries system with a user profile that has sufficient authority. See 2.10.1, “Authority requirement” on page 25 for details.
- Run either the setup script in QShell or the Run Java (RUNJVA) command.

2.11.1 Installing via the SETUP script in QShell

Use this option if you prefer a shorter command. Perform these steps to install WebSphere Application Server from Qshell using the SETUP script:

1. Place the WebSphere Application Server 5.0 for iSeries CD-ROM in the CD-ROM drive of your iSeries server.

   Note: Your user profile must have sufficient authority; see 2.10.1, “Authority requirement” on page 25 for details.

2. Enter STRQSH on an OS/400 command line to start the Qshell.
3. Enter `cd /QOPT/WEBSPHERE` to change directories to the root installation directory on the CD-ROM.
4. Run the SETUP script to start the installation program. The default is to install all product components (WebSphere Application Server, WebSphere MQ V5.3 for iSeries, and WebSphere MQ classes for Java and JMS V5.3 for iSeries) with all supported code options, and the primary language of the iSeries server if supported by the corresponding products or English 2924 if the primary language is not supported by the product. If you do not want to install all of the products, or you want to install a different language, you can specify optional parameters on the command line. For a list of the available parameters; see 2.11.3, “Parameters for SETUP script and RUNJAVA command” on page 28. To accept the default settings for the installation program, enter this command:

   ```
   SETUP
   ```

   Note: Do not issue any other commands, unless prompted, until the installation is completed. Doing so may cause the installation to stop prematurely.

5. Messages are displayed in QShell that indicate the progress of the installation process.

   Example 2-2 shows a partial sample of output from the installation process.

Example 2-2 Output from WebSphere Application Server 5.0 for iSeries installation process

```
> cd /QOPT/WEBSPHERE
$
> SETUP
Loading installation program. Please wait.
Checking current configuration. Please wait.
Installing selected options. Please wait.
Note: This may take up to 120 minutes to complete.
Product 5724B41: WebSphere MQ V5.3 for iSeries
(Code: Option Base)
   Copying stream file to save file.
   Restoring licensed program.
(Language: Option Base)
```

```
   Copying stream file to save file.
   Restoring licensed program.
(Code: Option 1)
   Copying stream file to save file.
   Restoring licensed program.
Product 5639C34: WebSphere MQ classes for Java and JMS V5.3 for iSeries
(Code: Option Base)
   Copying stream file to save file.
   Restoring licensed program.
(Language: Option Base)
   Copying stream file to save file.
   Restoring licensed program.
(Code: Option 1)
   Copying stream file to save file.
   Restoring licensed program.
Product 5733WS5: WebSphere Application Server V5.0 for iSeries
(Code: Option Base)
   Copying stream file to save file.
   Restoring licensed program.
(Language: Option Base)
   Copying stream file to save file.
   Restoring licensed program.
(Code: Option 1)
   Copying file to the iSeries.
```

6. When the installation completes, you should see the following line:

   ```
   Installation completed successfully.
   ```

7. Press F3 to quit QShell.

2.11.2 Installing via the Run Java (RUNJVA) command

The procedure is very similar to the SETUP script in Qshell. Use this command if you prefer control language (CL) commands. To install WebSphere Application Server to your iSeries using RUNJVA, perform these steps:

1. Place the WebSphere Application Server 5.0 for iSeries CD-ROM in the CD-ROM drive of your iSeries server.

Note: Your user profile must have sufficient authority. See 2.10.1, “Authority requirement” on page 25 for details.

The default installation will install all products (WebSphere Application Server, WebSphere MQ V5.3 for iSeries, and WebSphere MQ classes for Java and JMS V5.3 for iSeries) with all supported code options, and the primary language of the iSeries server if supported by the corresponding products, or English 2924 if the primary language is not supported by the product. If you do not want to install all of the products, or you want to install a different language, you can specify optional parameters on the command. For a list of the available parameters; see 2.11.3, “Parameters for SETUP script and RUNJAVA command” on page 28.

To accept the default settings for the installation program, enter this command, using the same capitalization as shown (this command has been wrapped for display purposes):

```
 RUNJVA CLASS(SETUP) CLASSPATH('/QIBM/ProdData/OS400/jt400/lib/jt400Native.jar:
/QOPT/WEBSPHERE/OS400:/tmp/WebSphere/WS5INSTALL.JAR') PROP((os400.runtime.exec QSHELL)
(java.version 1.3))
```

Note: If you type an error in this command, such as a wrong classpath, you will get an error message JVA0122, like this:

```
Java program completed with exit code 106
```

After you enter the RUNJVA command, messages in the command window indicate the progress of the installation process (see Example 2-2 on page 26).

2.11.3 Parameters for SETUP script and RUNJAVA command

The SETUP script in Qshell and the Run Java (RUNJVA) command include several optional parameters that you can use to customize your installation. If a parameter is not specified, the installation program uses the default value for that parameter. If no parameters are specified, WebSphere Application Server V5.0 for iSeries, WebSphere MQ V5.3 for iSeries, and WebSphere MQ classes for Java and JMS V5.3 for iSeries will be installed with their default options. For example, you can use parameters to install only the WebSphere Application Server product and skip WebSphere MQ V5.3 for iSeries and WebSphere MQ classes for Java and JMS V5.3 for iSeries. To do this, you would enter one of these commands to start the installation program:

- From Qshell:

  ```
  SETUP -wmq -skip true -wmqjava -skip true
  ```

- From an OS/400 command line:

  ```
  RUNJVA CLASS(SETUP) PARM('-wmq' '-skip' 'true' '-wmqjava' '-skip' 'true') CLASSPATH
  ('/QIBM/ProdData/OS400/jt400/lib/jt400Native.jar:/QOPT/WEBSPHERE/OS400:
  /tmp/WebSphere/WS5INSTALL.JAR') PROP((os400.runtime.exec QSHELL) (java.version 1.3))
  ```

  Here, the options specified within PARM() are the installation options.

  Note: This command has been wrapped for display purposes. Enter it on one line.

Table 2-4 describes all of the configurable parameters for the WebSphere Application Server, WebSphere MQ V5.3 for iSeries, and the WebSphere MQ classes for Java and JMS V5.3 for iSeries installation.

Important: If you decide to run selective install of WAS components, you may need to run additional commands. If you install WebSphere MQ for iSeries V5.3 (5724B41) after the WAS product (5733WS5 - Option 2) has already been installed, you will need to run the following commands after installing WebSphere MQ for iSeries V5.3:

- `QSYS/CHGUSRPRF USRPRF(QMQM) STATUS(*ENABLED) PWDEXPITV(*NOMAX)`

  This command changes the password expiration interval for the QMQM user profile to *NOMAX (it doesn't expire).

- `QSYS/CHGUSRPRF USRPRF(QMQMADM) STATUS(*ENABLED) PWDEXPITV(*NOMAX)`

  This command changes the password expiration interval for the QMQMADM user profile to *NOMAX (it doesn't expire).

- `QMQM/CHGMQMCAP CAPUNITS(*YES)`

  This command is used to indicate that the sufficient license units have been purchased for this installation of WebSphere MQ.

Table 2-4 Installation parameters

Product parameter name	Product assigned parameters	Possible values	Default	Description
-was	-skip	false true	**false**	If set to true, the WebSphere Application Server product is not installed.
	-language	2980 - Brazilian Portuguese 2950 - English Uppercase only 2924 - English 2938 - English Uppercase only DBCS 2894 - English DBCS 2928 - French 2940 - French MNCS 2929 - German 2939 - German MNCS 2932 - Italian 2942 - Italian MNCS 2962 - Japanese DBCS 2986 - Korean DBCS 2989 - Simplified Chinese DBCS 2931 - Spanish 2987 - Traditional Chinese DBCS	***PRIMARY, if available; otherwise 2924 (English)**	Determines the language to install. If the language parameter is not specified, the installation program detects the primary language on the iSeries server. If that language is supported by the WebSphere Application Server product, the program installs it. If the primary language is not supported, the installation program installs 2924 (English).
	-component	all code language	**all**	Selects which components to install. The code component installs only the code. The language component installs only the language options (29*nn*). The value *all* includes both the code and language components.
	-option	all base 1 2 3	**all**	Selects which options to install. The base option installs the common code. Option 1 installs the client code. Option 2 installs the server code. Option 3 installs the samples. The value all includes the options base, 1, 2, and 3. Installation of options 1, 2, and 3 requires that the base option is also selected or already installed. Note: You may specify the option parameter multiple times for more advanced configurations (for example, -option base -option 1 -option 3).

Product parameter name	Product assigned parameters	Possible values	Default	Description
-wmq	-skip	false true	**false**	If set to true, the WebSphere MQ product is not installed.
	-language	2909 - Belgium English 2966 - Belgium French 2981 - Canadian French MNCS 2950 - English Uppercase only 2924 - English 2938 - English Uppercase only DBCS 2984 - English DBCS 2928 - French 2940 - French MNCS 2932 - Italian 2942 - Italian MNCS 2962 - Japanese DBCS 2986 - Korean DBCS 2989 - Simplified Chinese DBCS 2931 - Spanish	***PRIMARY, if available; otherwise 2924 (English)**	Determines the language to install. If the language parameter is not specified, the installation program detects the primary language on the iSeries server. If that language is supported by WebSphere MQ, the program installs it. If the primary language is not supported, the installation program installs 2924 (English). Note: The language parameter only applies to the base option of WebSphere MQ. The other options are in English only.
	-component	all code language	**all**	Selects which components to install. The code component installs only code component. The language component installs only language options (29*nn*). The value all includes both the code and language components.
	-option	all base 1	**all**	Selects which options to install. The base option installs the server code Option 1 installs the samples code. The value all includes the options base and 1. Installation of option 1 requires that the base option is also selected or already installed.
-wmqjava	-skip	false true	**false**	If set to true, the WebSphere MQ classes for Java and JMS product is not installed.
	-component	all code language	**all**	Selects which component to install. The code component installs only the code. The language component install only the language option 2924. The value, all, includes both the code and language components. Note: The language parameter only applies to the base option of WebSphere MQ class for Java and JMS. The other options are in English only.

Product parameter name	Product assigned parameters	Possible values	Default	Description
	-option	all base 1	**all**	Selects which options to install. The base option installs the server code Option 1 installs the samples code. The value all includes the options base and 1. Installation of option 1 requires that the base option is also selected or already installed.

2.12 Installing WAS from the CD-ROM drive of your workstation

You can perform a remote installation of WebSphere Application Server from a Windows 32-bit operating system workstation. You can choose to install through one of these options:

- Abstract windowing toolkit (AWT):

  The AWT mode uses a graphical user interface (GUI) to complete the installation. The AWT mode is the default mode when you install WebSphere Application Server from a workstation.

- Silent:

  The silent mode allows us to enter a single command to start the installation. Silent installation is particularly useful if you install the product often or if you want to deploy the same configuration to a number of servers. The silent mode allows you to specify a response file which contains the installation option values you want to use. For any option not specified or commented out within the response file, the installer uses the default values.

2.12.1 Installing WAS from workstation using AWT mode

To use the AWT mode to install WAS iSeries from a Windows 32-bit workstation, perform these steps:

1. Verify that the host server jobs have been started on your iSeries server. The host server jobs allow the installation code to run on iSeries.

   Enter this command on an OS/400 command line:

   ```
   STRHOSTSVR SERVER(*ALL)
   ```

2. If TCP/IP is not started or if you don't know if TCP/IP is started, enter the Start TCP/IP (STRTCP) command on OS/400 command line.
3. Place the *WebSphere Application Server 5.0 for iSeries* CD-ROM in the CD-ROM drive of the workstation.

Note: Do not use the IBM WebSphere Application Server Version 5 for Windows NT and Windows 2000 CD-ROM (which was shipped with your WebSphere Application Server for iSeries package) for this set of steps.

The InstallShield program should automatically start. If it does not, open Windows Explorer and select your CD-ROM drive. Double-click the SETUP.EXE file to start the InstallShield program.

4. At the first panel, read the information and click **Next**.
5. On the next panel, enter the name of the iSeries server where you are installing WebSphere Application Server. You also must enter a valid user ID and password for the server. This user ID must have sufficient authority. See 2.10.1, “Authority requirement” on page 25 for details.
6. Click **Next**.
7. On the next panel, select the installation options for WebSphere Application Server V5.0 for iSeries (see Figure 2-7). Refer to 2.1, “What products and options are included?” on page 12 for the details.

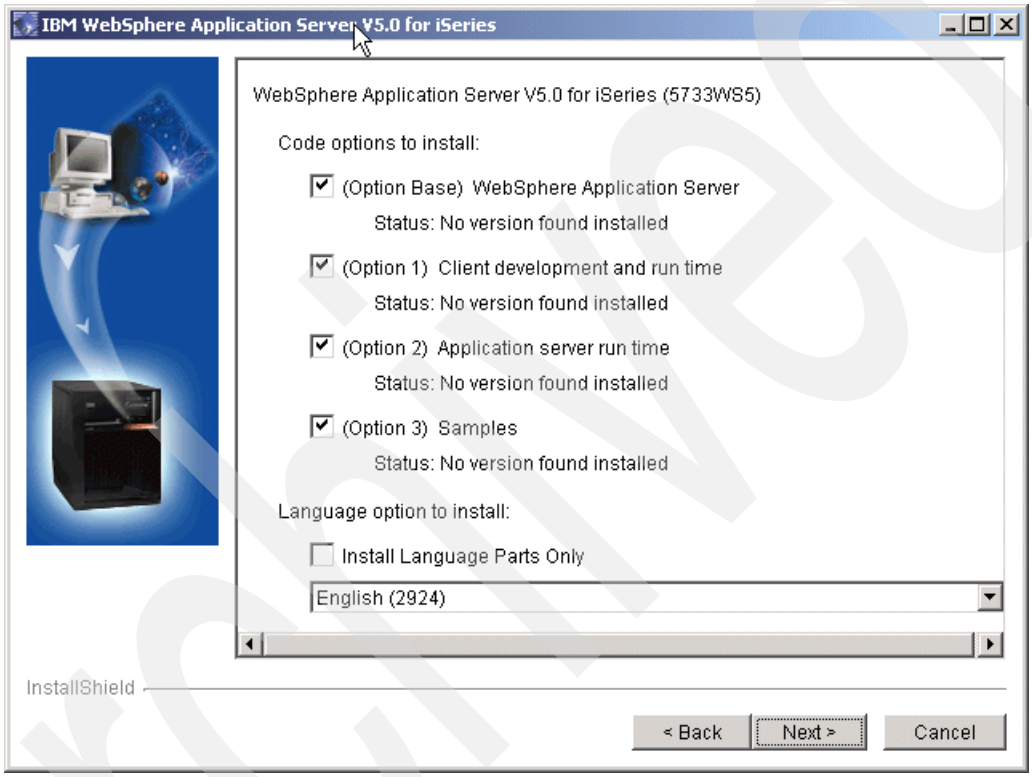


Figure 2-7 Select option of WebSphere to install

8. Click **Next**.

   On the next panel, select the installation options for WebSphere MQ V5.3 for iSeries (see Figure 2-8). Refer to 2.1, “What products and options are included?” on page 12 for the details.

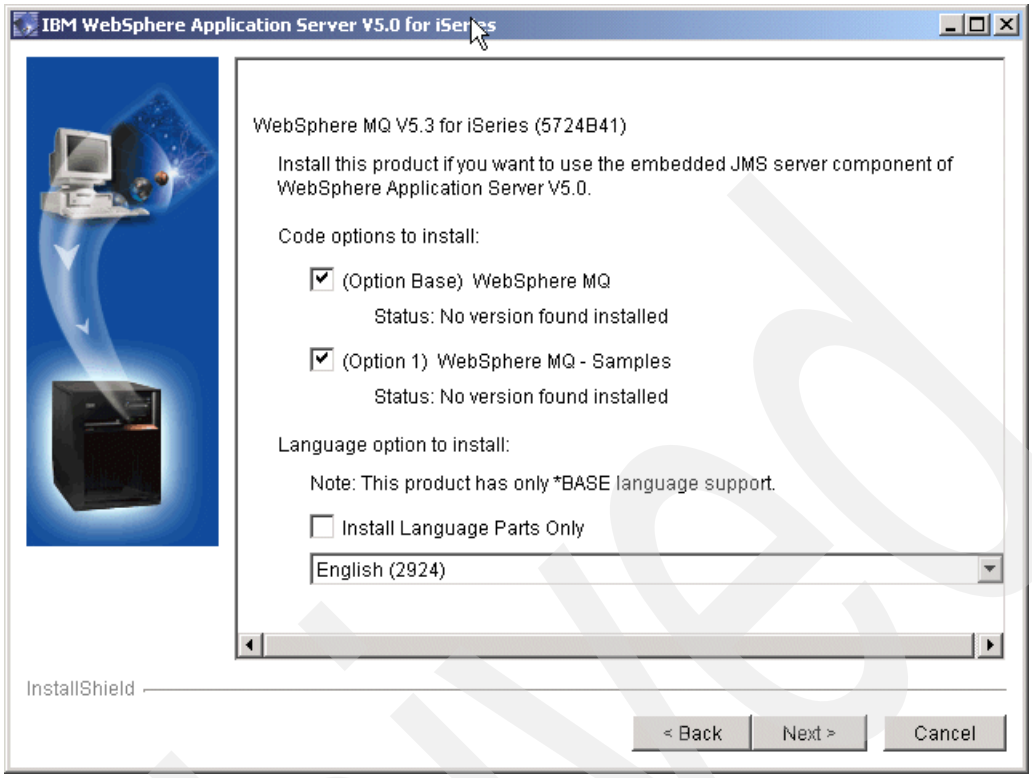


Figure 2-8 Select MQ option to install

9. Click **Next**.
10. On the next panel select the installation options for WebSphere MQ classes for Java and JMS V5.3 for iSeries (see Figure 2-9). Refer to 2.1, “What products and options are included?” on page 12 for the details.

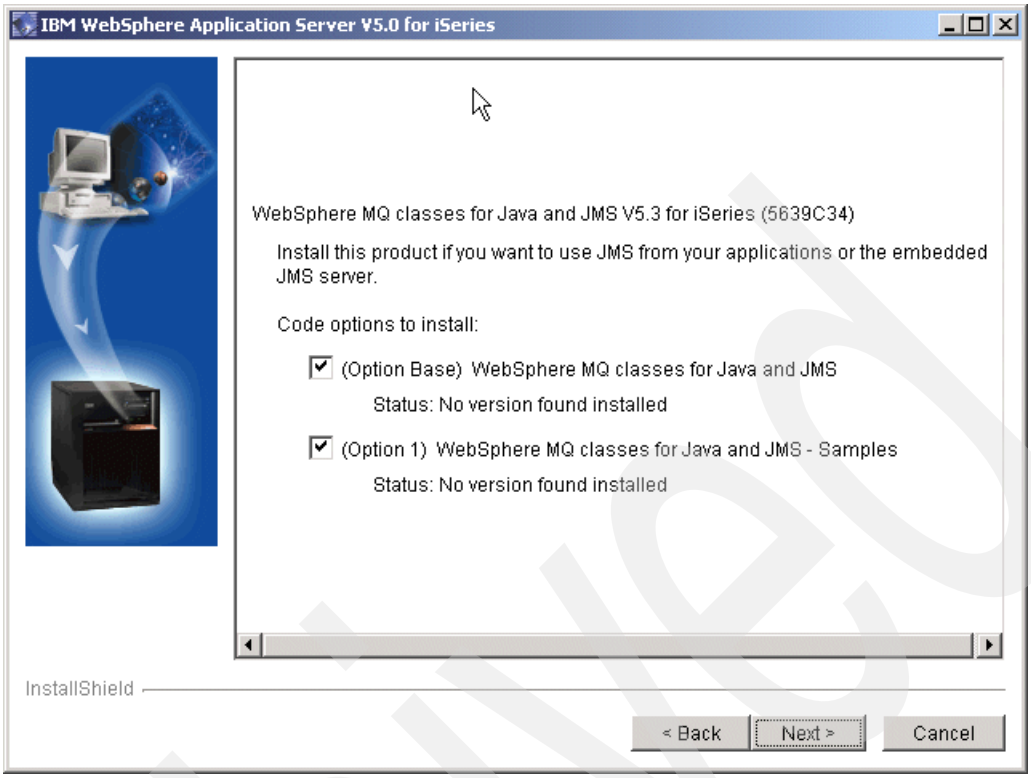


Figure 2-9 Select options to install MQ classes for Java and JMS V5.3 for iSeries:

11. Click **Next**.
12. The summary screen for the selected products and options is displayed (see Figure 2-10). If they are not correct, click **Back** to change your installation options.

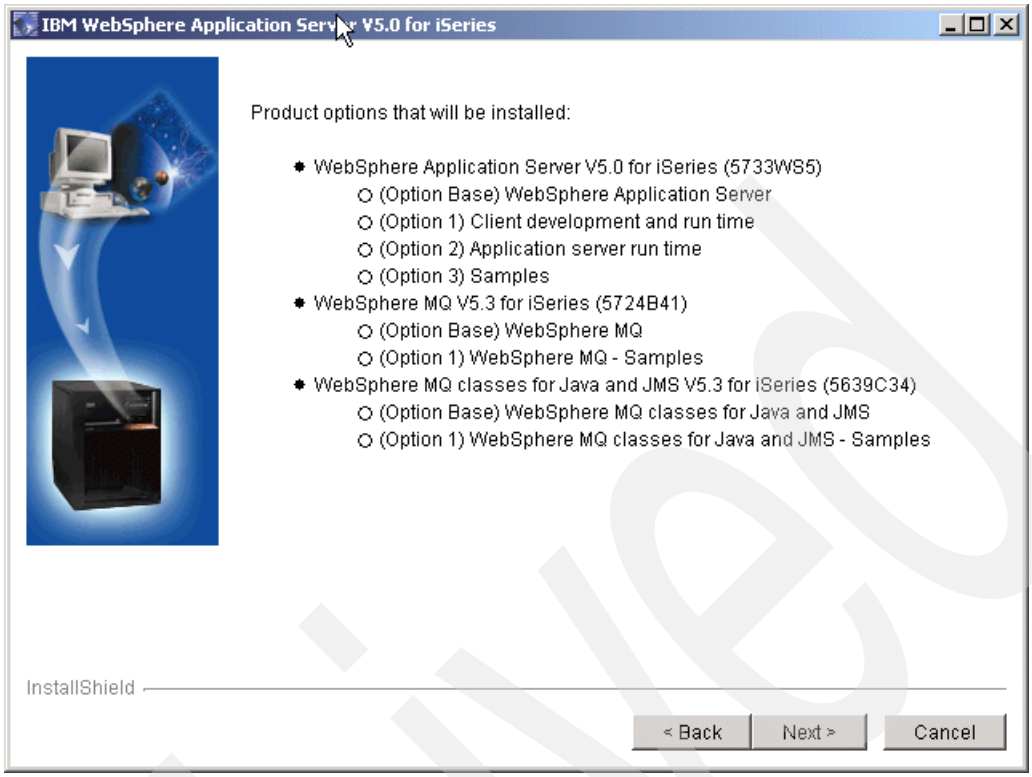


Figure 2-10 confirm options that you specified

13. If they are correct, click **Next**.

    The InstallShield program displays messages that indicate the status of the installation and a status bar to show the progress of the installation.

Important: When you start the installation, a command window is opened in the background. InstallShield logs all messages to this command window. The messages are very similar to the ones shown in Example 2-2 on page 26.

The message to insert the second CD will be displayed in the command window.

14. After the installation is complete, the summary panel is displayed showing the options that were installed.
15. Click **Finish** to close the InstallShield program.
16. For security purposes, if the host servers were not running, you should return your iSeries server to its original state with the End Host Server (ENDHOSTSVR) command after the install is complete.

2.12.2 Installing WebSphere Application Server using silent mode

If you want to install WebSphere Application Server 5.0 for iSeries with the same parameters onto a number of servers, silent mode installation is a good choice. The silent mode allows you to install the product with a single command. Installation options must be specified in a response file. During the installation, you are unable to change the installation options. The parameters and default values are discussed in 2.12.3, “Parameters for silent installations of WebSphere Application Server” on page 37.

To use the silent mode, follow these steps:

1. Verify that the host server jobs have been started on your iSeries server. The host server jobs allow the installation code to run on iSeries.

   Enter this command on an OS/400 command line:

   ```
   STRHOSTSVR SERVER(*ALL)
   ```

2. If TCP/IP is not started, enter the Start TCP/IP (STRTCP) command on OS/400 command line.
3. Place the *WebSphere Application Server 5.0 for iSeries* CD-ROM in the CD-ROM drive of your workstation. The autorun feature will bring up the GUI; click **Cancel** to exit the GUI.

   Note: Your user profile must have sufficient authority. See 2.10.1, “Authority requirement” on page 25 for details.

4. On your workstation, open a command window.
5. Access the CD-ROM drive of your workstation by switching to the CD-ROM drive. For example, enter e: where e: is the letter assigned to your CD-ROM drive.
6. Copy the RESPONSEFILE file from the CD-ROM directory to a directory on your workstation such as c:\temp. For example:

   ```
   copy RESPONSEFILE c:\temp\RESPONSEFILEWAS5B.txt
   ```

7. Edit the file using an editor of your choice and modify the installation options to suit your installation requirements (see 2.12.3, “Parameters for silent installations of WebSphere Application Server” on page 37).
8. Enter the SETUP command, specifying the response file to be used during the installation. For example:

   ```
   SETUP -options c:\temp\RESPONSEFILEWAS5B.txt
   ```

9. When the installation completes, you should see the following message:

   ```
   Installation completed successfully.
   ```

10. If you do something wrong on the silent install script, an error message as shown in Example 2-3 will be prompted:

Example 2-3 possible error in Silent install

```
Checking current configuration.  Please wait.
Could not establish a connection to the iSeries.  java.net.UnknownHostException:
 YOURSYSTEM
Installation failed, check previous messages.

Please read the Installation and Initial Configuration documentation:
  http://www.ibm.com/websphere/iseries/installws5

Please press the Enter key to end the installation program.
```

11. When the installation is complete, press any key to exit the installation program.
12. For security purposes, if the host servers were not running, you should return your iSeries server to its original state with the End Host Server (ENDHOSTSVR) command after the install is complete.

2.12.3 Parameters for silent installations of WebSphere Application Server

When you install WebSphere Application Server to your iSeries server from a workstation using the silent mode, you refer to a response file that specifies a silent install. To configure the installation, change the options in this file before you enter the installation command (SETUP). The silent installation mode does not accept interactive installation options. If you want to specify non-default installation options, you must use the response file.

Example 2-4 shows a sample response file. By using this file with the SETUP command, all products with all options will be installed.

Example 2-4 Sample silent install response file

```
############################################################################
#                                                                          #
# ABOUT:          WebSphere Application Server V5.0 for iSeries            #
#                                                                          #
# INSTALL TYPE:   Remote workstation silent installation                   #
#                                                                          #
# DOCUMENTATION:  Please see the WebSphere Application Server              #
#                 for iSeries Installation and Initial Configuration       #
#                 Web site for the most up-to-date information:            #
#                                                                          #
#                 http://www.ibm.com/websphere/iseries/installws5          #
#                                                                          #
# USAGE:          SETUP -options <path>\<responsefile>                     #
#                                                                          #
# IMPORTANT:      All values must be enclosed in double quotes.            #
#                                                                          #
############################################################################

# Specify silent install
-silent

# Bean property assignments for target iSeries system
-W DestinationBean.system="RCHASO2B"
-W DestinationBean.username="PONGACH"
-W DestinationBean.password="MAYPONG"

# Bean property assignments for WAS
# Note: option property can be specified more than once
-W WASSelectBean.skip="false"
# -W WASSelectBean.language="29XX"
-W WASSelectBean.component="all"
-W WASSelectBean.option="all"

# Bean property assignments for WebSphere MQ
# Note: option property can be specified more than once
-W WMQSelectBean.skip="false"
# -W WMQSelectBean.language="29XX"
-W WMQSelectBean.component="all"
-W WMQSelectBean.option="all"

# Bean property assignments for WebSphere MQ Java
-W WMQJavaSelectBean.skip="false"
-W WMQJavaSelectBean.option="all"
```

Note: There are several points to be aware of:

1. All the usernames and passwords in the response file are in clear text format. You need to beware of any security exposure.
2. The response file is in English only.
3. Install options preceded by # are commented out. To enable these options, remove the # and the space.
4. If you want the default value, you do not need to specify it.
5. The DestinationBean property must be edited.

You can use Table 2-5 to select the possible parameters for the WebSphere Application Server for remote silent install response file. The table also lists the supported national language options including double-byte character sets (DBCS) and multi-national character sets (MNCS).

Table 2-5 Possible parameters for remote silent install response file

Bean ID	Bean property	Possible values	Default	Description
DestinationBean	System			Determines the destination iSeries server upon which the installation is to be performed.
	username			Determines the user name to be used on the destination iSeries server.
	password			Determines the password to be used on the destination iSeries server.

Bean ID	Bean property	Possible values	Default	Description
WASSelectBean	skip	**false** true	false	If set to true, the WebSphere Application Server product is not installed.
	language	2980 - Brazilian Portuguese 2950 - English Uppercase only 2924 - English 2938 - English Uppercase only DBCS 2894 - English DBCS 2928 - French 2940 - French MNCS 2929 - German 2939 - German MNCS 2932 - Italian 2942 - Italian MNCS 2962 - Japanese DBCS 2986 - Korean DBCS 2989 - Simplified Chinese DBCS 2931 - Spanish 2987 - Traditional Chinese DBCS	*PRIMARY, if available; otherwise 2924 (English)	Determines the language to install. If the language parameter is not specified, the installation program detects the primary language on the iSeries server. If that language is supported by the WebSphere Application Server product, the program installs it. If the primary language is not supported, the installation program installs 2924 (English).
	component	**all** code language	all	Selects which components to install. The code component installs only the code. The language component installs only the language options (29nn). The value all includes both the code and language components.
	option	**all** base 1 2 3		Selects which options to install. The base option installs the common code. Option 1 installs the client code. Option 2 installs the server code. Option 3 installs the samples. The value all includes the options base, 1, 2, and 3. Installation of options 1, 2, and 3 requires that the base option is also selected or already installed. Note: You may specify the option parameter multiple times for more advanced configurations. For example -W WASSelectBean.option = "base" -W WASSelectBean.option = "1" -W WASSelectBean.option = "3").

Bean ID	Bean property	Possible values	Default	Description
WMQSelectBean	skip	**false** true	false	If set to true, the WebSphere MQ product is not installed.
	language	2909 - Belgium English 2966 - Belgium French 2981 - Canadian French MNCS 2950 - English Uppercase only 2924 - English 2938 - English Uppercase only DBCS 2984 - English DBCS 2928 - French 2940 - French MNCS 2932 - Italian 2942 - Italian MNCS 2962 - Japanese DBCS 2986 - Korean DBCS 2989 - Simplified Chinese DBCS 2931 - Spanish	*PRIMARY, if available; otherwise 2924 (English)	Determines the language to install. If the language parameter is not specified, the installation program detects the primary language on the iSeries server. If that language is supported by WebSphere MQ V5.3 for iSeries, the program installs it. If the primary language is not supported, the installation program installs 2924 (English). Note: The language parameter only applies to the base option of WebSphere MQ V5.3 for iSeries. The other options are in English only.
	component	**all** code language	all	Selects which components to install. The code component installs only the code. The language component installs only the language options (29nn). The value all includes both the code and language components.
	option	**all** base 1	all	Selects which options to install. The base option installs the server code. Option 1 installs the samples code. The value all includes the options base and 1. Installation of option 1 requires that the base option is also selected or already installed.
	skip	**false** true	false	If set to true, the WebSphere MQ classes for Java and JMS product is not installed.

Bean ID	Bean property	Possible values	Default	Description
	component	**all** code language	all	Selects which components to install. The code component installs only the code. The language component installs only the language option (2924). The value all includes both the code and language components. Note: The language parameter only applies to the base option of WebSphere MQ class for Java and JMS. The other options are in English only.
	option	**all** base 1	all	Selects which options to install. The base option installs the server code. Option 1 installs the samples code. The value all includes the options base and 1.

2.13 Installed Licence Programs

Figure 2-11 shows the installed licence programs display when you have installed all options of the WebSphere Application server. Use the DSPSFWRSC command and scroll in the display shown for the different components.

```
                         Display Software Resources
                                                              System:
RCHAS02B
Resource                  Feature
   ID     Option  Feature   Type    Library     Release
5733WS5   *BASE    5050    WebSphere Application Server V5.0
5733WS5   *BASE    2924    WebSphere Application Server V5.0
5733WS5   1        5050    WAS V5.0 Client development and runtime
5733WS5   1        2924    WAS V5.0 Client development and runtime
5733WS5   2        5050    WAS V5.0 Application server runtime
5733WS5   2        2924    WAS V5.0 Application server runtime
5733WS5   3        5050    WAS V5.0 Samples
5733WS5   3        2924    WAS V5.0 Samples
5724B41   *BASE    5050     *CODE   QMQM        V5R3M0
5724B41   *BASE    2924     *LNG    QMQM        V5R3M0
5724B41   1        5050     *CODE   QMQMSAMP    V5R3M0
5639C34   *BASE   5050    WebSphere MQ classes for Java and JMS - V5.3
5639C34   1        5050    WebSphere MQ Classes for Java - samples
```

Figure 2-11 Sample DSPSFWRSC screen after installation of WAS

2.14 Installing WebSphere group PTF

All product prerequisites must be installed before you install the group PTF package, or WebSphere Application Server may fail when it is started. For example, the Java PTFs contained in the package will not be installed if IBM Developer Kit for Java 1.3 (5722-JV1, Option 5) is not installed on the server.

These instructions describe how to install the WebSphere Application Server for iSeries group PTF:

1. Verify that all of the prerequisite software is installed. You can check for these in 2.2.2, “iSeries and AS/400 software requirements” on page 14.
2. Place the WebSphere for iSeries group PTF CD-ROM into the CD-ROM drive on your iSeries server.
3. Sign on to your iSeries server. Your user profile must have sufficient authority (see 2.10.1, “Authority requirement” on page 25 for details).
4. Enter this command to bring your system into a restricted state:

   ```
   ENDSBS SBS(*ALL)
   ```

5. Enter this command from an OS/400 command line when the system is in a restricted state:

   ```
   GO PTF
   ```

6. Select option 8 (Install program temporary fix package) from the menu.
7. Specify these values and press Enter (see Figure 2-12):
   a. **Device:** (Specify the device name of your CD-ROM drive, for example, OPT01.)
   b. **Automatic IPL:** Y
   c. **PTF type:** 1 (All PTFs)

```
                    Install Options for Program Temporary Fixes
                                          System: your.server
Type choices, press Enter.

Device . . . . . .          OPT01     Name, *SERVICE
Automatic IPL . . . . . .   Y         Y=Yes
                                      N=No
Restart type . . . . . .    *SYS      *SYS, *FULL
PTF type . . . . . .        1         1=All PTFs
                                      2=HIPER PTFs and HIPER
                                          LIC fixes only
                                      3=HIPER LIC fixes only
                                      4=Refreshed Licensed Internal Code
Other options . . . . . .   N         Y=Yes
                                      N=No
F3=Exit  F12=Cancel
```

Figure 2-12 Install program temporary fix package

After all of the PTFs have been installed, your iSeries server restarts.

Starting from OS/400 V5R2, you can make use of Work with PTF Groups (WRKPTFGRP) to check which group PTFs are installed on your iSeries server.

Figure 2-13 shows the group PTFs on the ITSO test system.

```
                                Work with PTF Groups
                                                              System:   RCHAS02B
Type options, press Enter.
  4=Delete   5=Display   6=Print   9=Display related PTF groups

Opt  PTF Group             Level  Status
     SF99502                   4  Installed
     SF99245                   1  Installed
     SF99169                   4  Installed
     SF99148                   3  Installed
     SF99098                   7  Installed









                                                                       Bottom
F3=Exit   F6=Print   F11=Display descriptions   F12=Cancel
F22=Display entire field
```

Figure 2-13 Work with PTF Groups (WRKPTFGRP) from V5R2

2.15 Verifying TCP/IP configuration

The IPTest Java utility is shipped with the WebSphere Application Server product and can be used to debug TCP/IP configuration problems. To run this utility, enter this command on an OS/400 command line:

```
RUNJVA CLASS(IPTest) CLASSPATH('/QIBM/ProdData/WebAS5/Base/bin')
```

Note: The class name parameter that you specify for CLASS (IPTest) is case-sensitive.

Figure 2-14 illustrates the output from this command.

```
                                Java Shell Display

 Local Address: 1.2.3.4
Local Name: RCHAS02B.IBM.COM
All addresses for RCHAS02B.IBM.COM:
      1.2.3.43
Java program completed


===>



F3=Exit  F6=Print F9=Retrieve F12=Exit
F13=Clear F17=Top  F18=Bottom  F21=CL command entry
```

Figure 2-14 Verifying by IPTest Java utility

Local Address is the IP address for your iSeries server, and **Local Name** is the domain-qualified host name for your iSeries server. This value must not be blank and must match the values you defined in the TCP/IP configuration; see “Starting, configuring, and verifying TCP/IP” on page 21. Press F3 to exit.

Note: If a host name has not been configured for your iSeries server, you will receive an UnknownHostException message.

2.16 Troubleshooting the installation

WebSphere Application Server offers several methods you can use to troubleshoot problems. Which method you use depends on the nature of the problem. Generally, you use a combination of several methods to determine the cause of a problem and then decide on an appropriate method for its resolution. For more information, refer to Chapter 11, “Troubleshooting” on page 431.

2.17 The library, subsystem, and job structure of WAS V5.0

This section describes the product library and subsystems that WebSphere Application Server use on the iSeries server. Having an understanding of this structure can help you to facilitate the administration, development, and troubleshooting tasks on WebSphere Application Server 5.0 for iSeries.

2.17.1 Product library and WAS subsystem

WebSphere Application Server 5.0 for iSeries uses the following library and subsystem:

- **QEJBAS5:** This is the product library.

WebSphere Application Server for iSeries uses these subsystems:

- **QEJBAS5 subsystem:** This is the WebSphere Application Server subsystem. For each WebSphere Application Server instance the application server job and depending of the configuration of the instance additional jobs like the JMS listener and the node agent run in this subsystem.
- **QMQM subsystem:** If your instance uses the embedded JMS server, additional jobs are started on your iSeries server when you start the instance. These jobs run in the QMQM subsystem.

2.17.2 WebSphere Application Server job

WAS jobs run in the QEJBAS5 subsystem. The job name is the first 10 characters of the application server name. If the first 10 characters do not provide a valid iSeries job name, the WebSphere Application Server runtime creates a valid job name. If the runtime cannot create a valid job name for the application server, it uses the default job name QEJBSVR. The default instance is configured with a single application server named **server1**, so the iSeries job name is *SERVER1*.

For more information about the WAS application server jobs, refer to our examples in 6.5.4, “Verifying that the WAS environment has started” on page 119.

2.17.3 MQ listener job

If the WAS instance is configured to use the embedded JMS server, the MQ listener job is used by the embedded JMS provider to establish connections to the embedded JMS server. The job name is *QEJBMQLSR*. If you are using multiple instances of WebSphere Application Server with embedded JMS enabled, you will see a number of the *QEJBMQLSR* jobs: one for each instance.

Note: The default WAS instance is *not* JMS-enabled.

To determine which QEJBMQLSR job is in use by your specific application server, you can display the joblog for your application server job to see the job information for the QEJBMQLSR job the application server is using. For an example showing how to do this, refer to 6.5.4, "Verifying that the WAS environment has started" on page 119.

For more information, refer to 6.13.3, "JMS Administration in WebSphere 5.0" on page 220.

2.17.4 Other jobs for embedded JMS

In addition to the jobs that run in the QEJBAS5 subsystem, WAS instances use other jobs. If the instance is configured to use the embedded JMS server, these jobs are started in the QMQM subsystem when your application server or node agent server is started:

```
AMQALMPX
AMQPCSEA
AMQRMPPA
AMQRRMFA
AMQZDMAA
AMQZLAAO
AMQZXMAO
RUNMQCHI
```

The joblog for each of these jobs contains a message indicating the queue manager name for the listener. You will see a message that is similar to the following one:

```
WebSphere MQ job 792 started for WAS_RCHAS07_jmsmdb_jmsmdbSvr.
```

The message contains the node name and the last two parts indicate the instance name and the application server name respectively. In our example:

- Node name: RCHAS07
- Instance name: jmsmdb
- Server name: jmsmdbSvr

2.17.5 The node agent job

The node agent job exists only when the instance is managed by a Deployment Manager (in other words, it is part of a cell). The job name for the node agent process is:

```
NODEAGENT
```

It runs in the QEJBAS5 subsystem. If you have added multiple nodes to your cell, you should see one NODAGENT job for each added node.

You can determine which NODEAGENT job is for a specific node as described in "Verifying that the WebSphere Application Server node agent has started" on page 292.

You can also find some information about the node agent job in 4.16, “IFS directory structure of WAS-ND” on page 82.

2.18 IFS directory structure

WebSphere Application Server V5.0 for iSeries uses the integrated file system (IFS) of the iSeries to store product, configuration and WAS application data. The following directories are used:

1. **/QIBM/ProdData/WebAS5/Base:** Root directory for the WebSphere Application Server product. Product data that is shared by all WebSphere Application Server instances is stored here. Additionally, the master copies of the files which make up an instance are stored here. Files under this directory structure should not be modified.
2. **/QIBM/ProdData/WebAS5/wemps:** Root directory for WebSphere Embedded Messaging Publish and Subscribe (embedded JMS). Never modify files under this directory structure.

   Important: The `/QIBM/ProdData/*` is the production directory. You must keep it intact without any modification. All the customizing is kept in the `/QIBM/UserData/*` directory.

3. **/QIBM/UserData/WebAS5/Base:** Root directory for all WAS instances. For every WAS instance you create, a new subtree is added under the `/QIBM/UserData/WebAS5/Base` directory (see Figure 2-15 on page 46). The name of the root directory for this subtree reflects the name of the WAS instance. The *default* directory is created during the installation.
4. **/QIBM/UserData/WebAS5/wemps:** Root directory for WebSphere Embedded Messaging Publish and Subscribe configuration and log files. The files under this directory are created and managed by the embedded JMS server, and should not be modified. See also 2.20, “Directories for JMS enabled instances” on page 50.

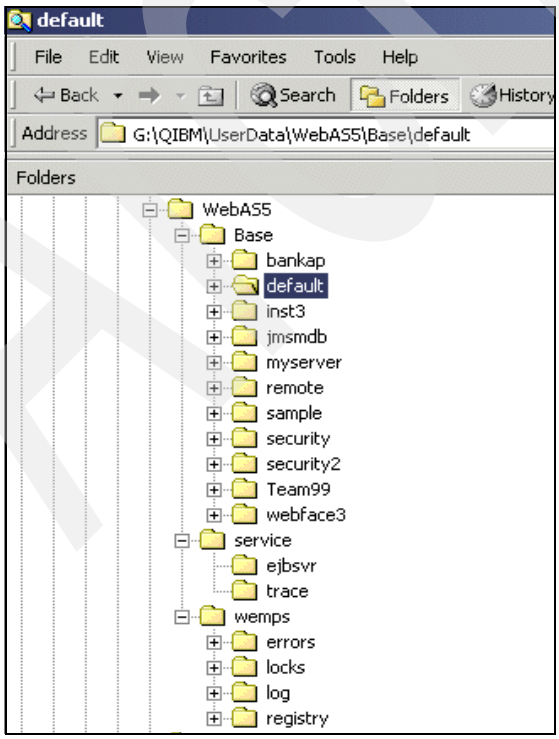


Figure 2-15 The IFS directory structure

2.19 WAS instance directory structure

For every WAS instance, you will find the same directory structure under the directory /QIBM/UserData/WebAS5/Base/*instance_name*. Figure 2-16 shows a sample directory structure of the *default* WAS instance.

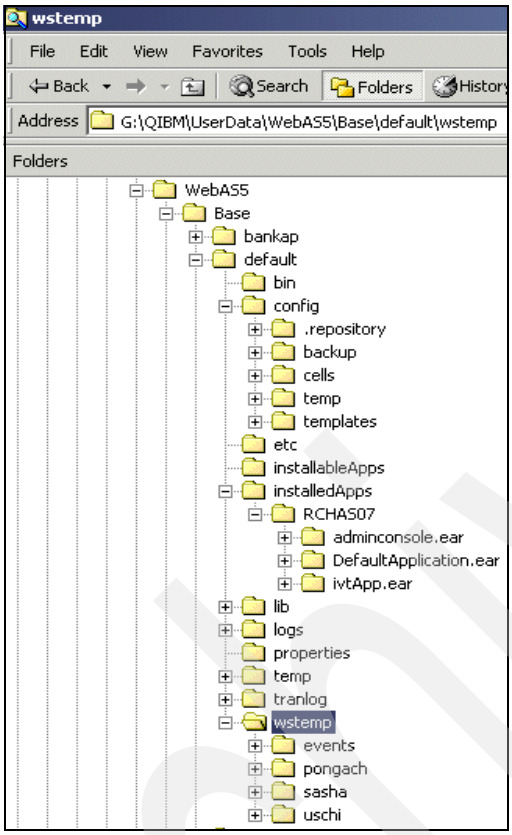


Figure 2-16 Directory structure on an instance

This section describes the WAS instance directory structure, emphasizing the significance of several subdirectories and files.

2.19.1 The *config* subdirectory

The config subdirectory contains a collection of the XML documents describing the configuration parameters of the WAS instance. This is a replacement of the DB2-based configuration repository for WebSphere Application Server V4.

Figure 2-17 shows a sample structure of the config subdirectory. All configuration data is split into 3 levels:

- Cell level
- Node level
- Server level

The directory structure reflects this separation (see Figure 2-17).

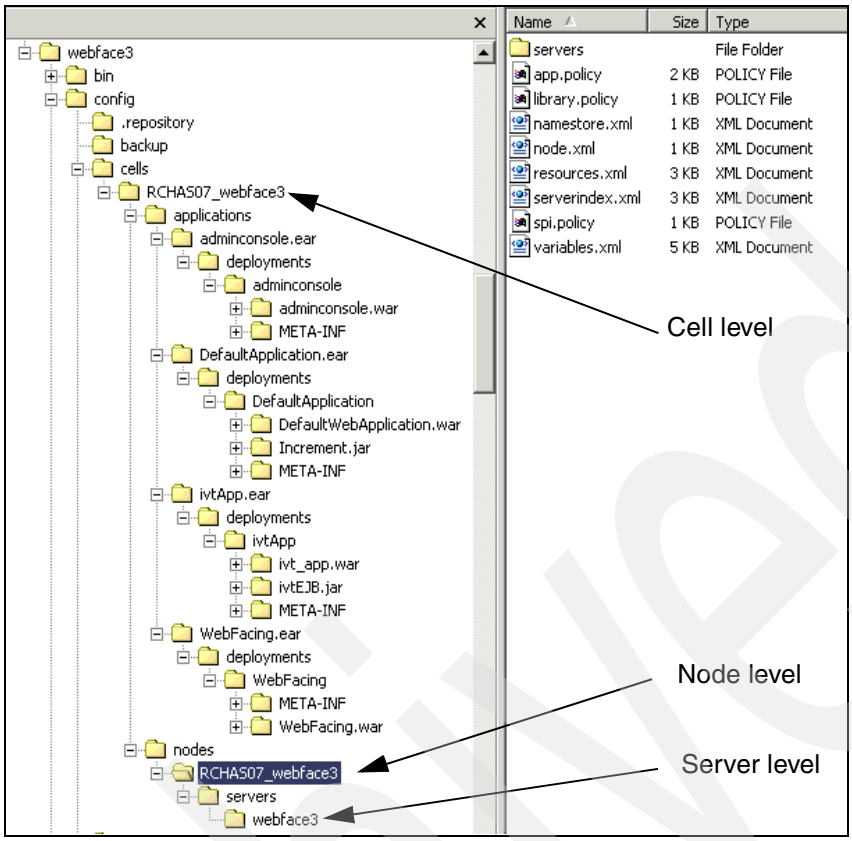


Figure 2-17 Configuration repository directories - example instance webface3

Important: Unless you are absolutely sure of what you are doing, DO NOT attempt to modify the XML files in the config subdirectories using a text editor: WAS is very sensitive to the format of the XML files. An incorrectly placed blank character can result in the failure of WAS to work properly.

Use one of the available administrative tools (the administrative console or the wsadmin tool) to modify the configuration parameters.

2.19.2 The *logs* subdirectory

As shown in Figure 2-18, for every instance there is a *logs* directory under the directory /QIBM/UserData/WebAS5/Base/instance_name. Here you can find the different log files that are produced, for example, the SystemErr.log and SystemOut.log. For more information, refer to 11.4, “WebSphere Application Server log files” on page 461.

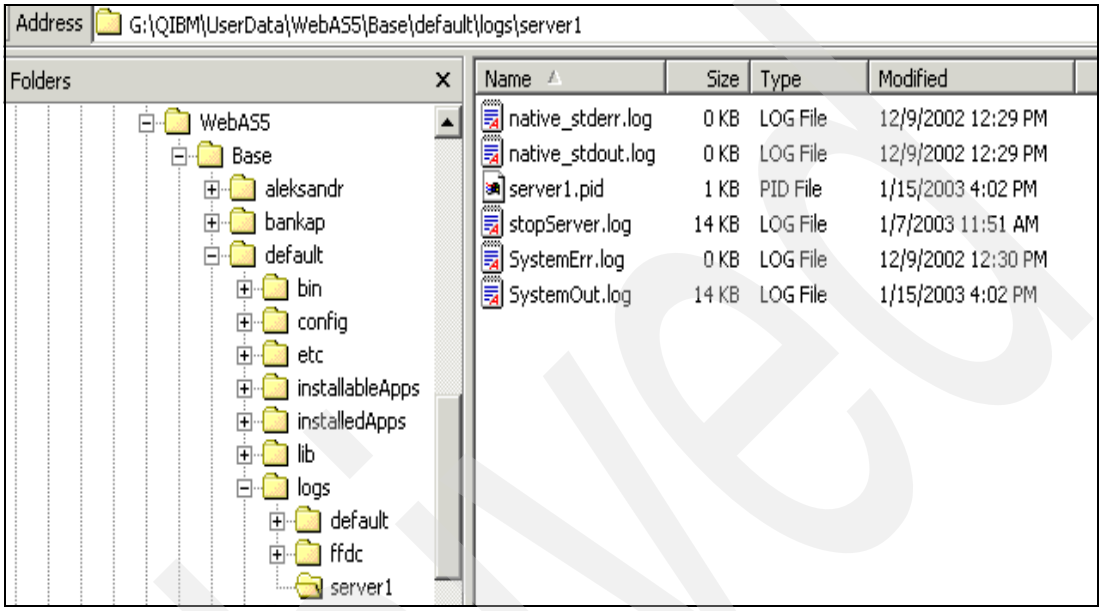


Figure 2-18 Log directory

2.19.3 The wstemp directory

When you perform administrative tasks with the administrative console, WebSphere Application Server generates temporary configuration files and backup configuration files in this directory. For an example, see Figure 2-19.

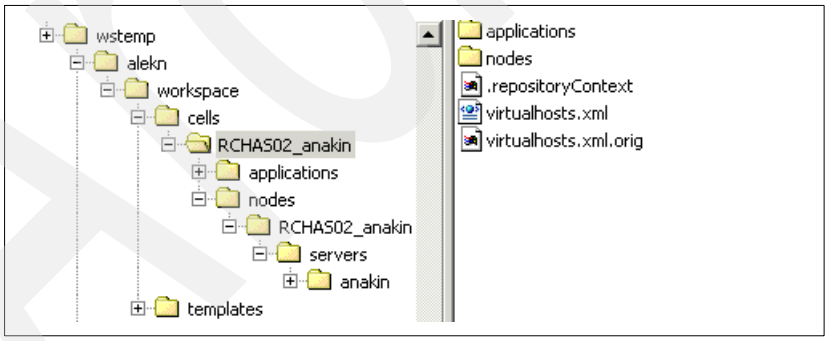


Figure 2-19 Example of the wstemp directory

In this example, we started the administrative console and changed the virtual host parameters. You can see the virtualhosts.xml file at the cell level. When a user commits the changes, the configuration files from the wstemp directory (in our example, virtualhosts.xml) are copied into the appropriate location under the instance’s /config directory.

2.20 Directories for JMS enabled instances

The `/QIBM/UserData/WebAS5/wemps` directory contains the configuration and other data for the JMS-enabled instances (see Figure 2-20). You can see a separate subdirectory for each JMS-enabled instance under the *registry* directory.

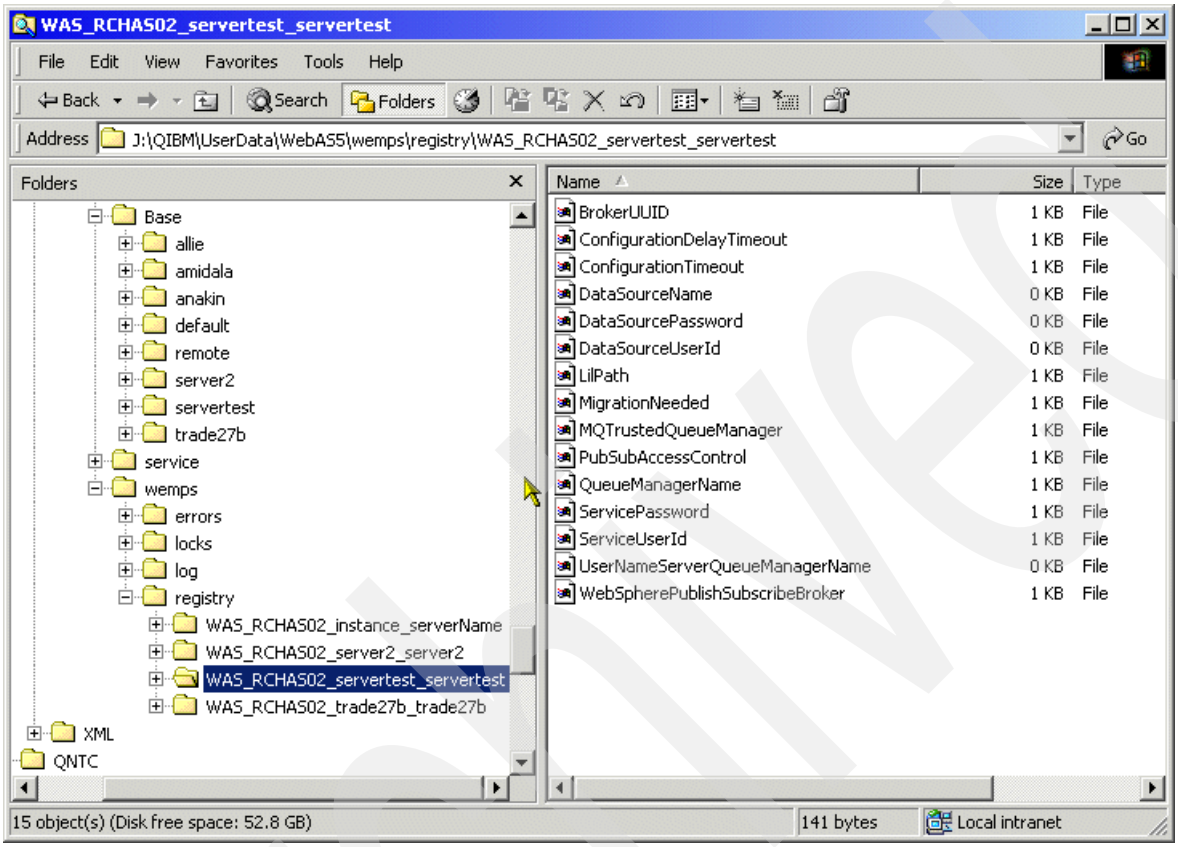


Figure 2-20 wemps directory structure

Also, for JMS enabled instances, the `/QIBM/UserData/mqm` directory holds information in an instance specific subdirectory as shown in Figure 2-21. For this instance, we use the embedded JMS support in the WebSphere application server.

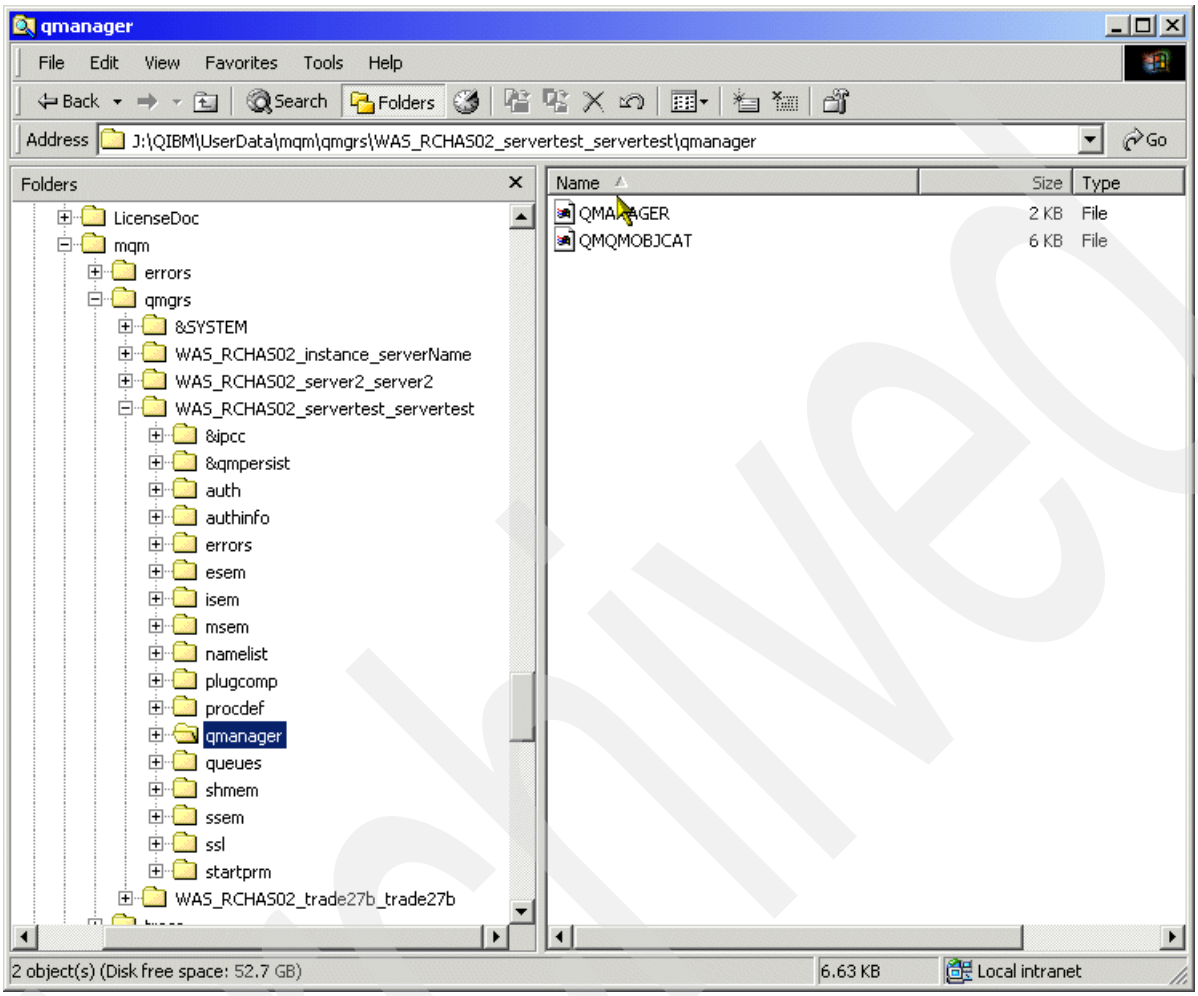


Figure 2-21 The /QIBM/UserData/mqm directory

2.21 Application data repository

Application data is stored in two different subdirectories:

- `/installedApps/*`
- `/config/cells/cellname/applications/*` (the configuration repository)

The `/installedApps/*` directory contains only the application binaries for applications that have modules deployed to an application server, for example JARs, class files, and JSPs.

The `config/cells/cellname/applications/*` directory contains the J2EE application deployment descriptor file (application.xml) as well as the IBM bindings and extensions files.

Figure 2-22 shows an example of the two subdirectories for the *ivtApp* application.

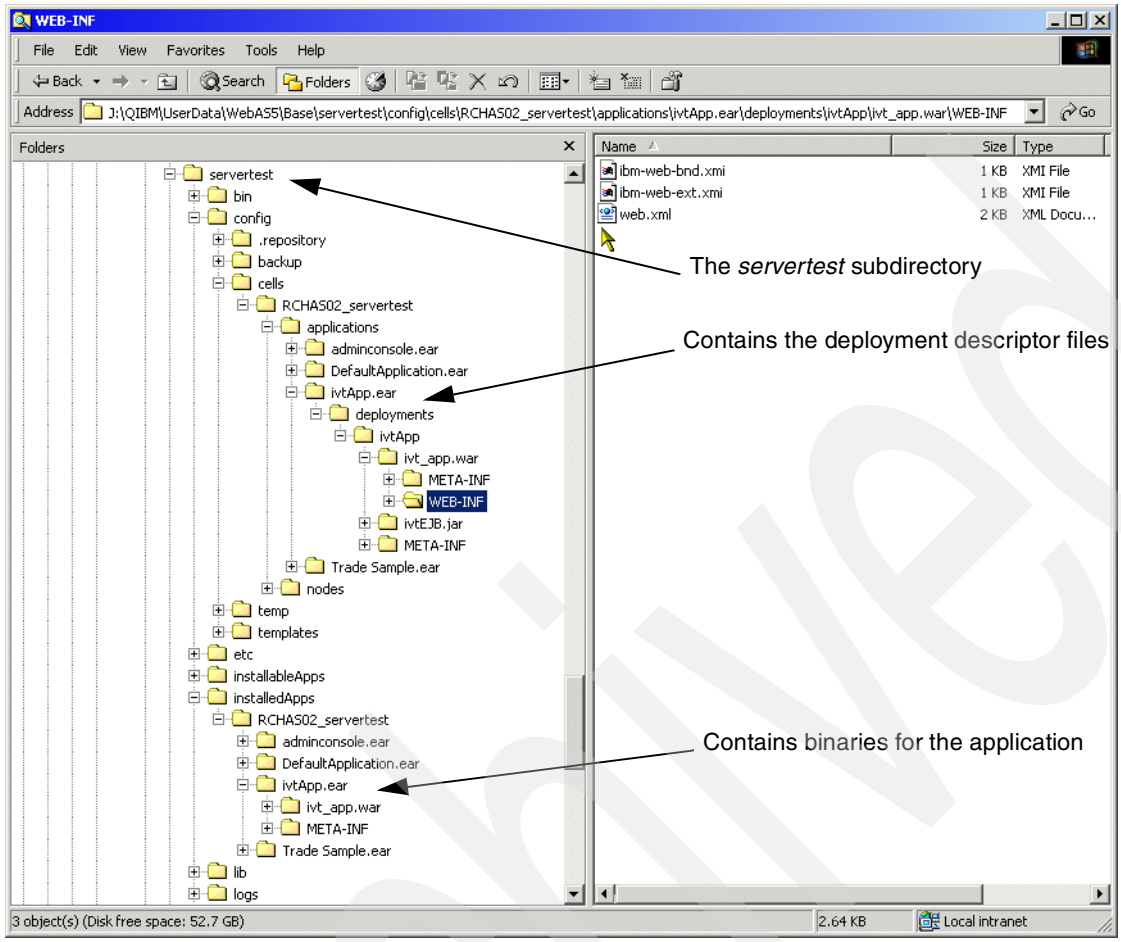


Figure 2-22 The applications subdirectories

2.22 Variable-scoped documents

Identically named documents that exist at differing levels of the configuration hierarchy are called *variable-scoped* documents. There are two uses for variable-scoped documents:

- In the first possibility, configuration data contained in a document at one level is logically combined with data from documents at other levels of the configuration hierarchy. In the case of conflicting definitions, the “most specific” value takes precedence. For example:, if an identical entry exists in the files at the cell and node level (like a variable defined in both the cell’s and node’s *variables.xml*), the entry at the node level takes precedence.
- The second possibility involves documents representing data that is not merged but is rather scoped to a specific level of the topology. For example: The *namestore.xml* document at the cell level contains the cell persistent portion of the namespace, while the *namestore.xml* at the node level contains the node persistent root of the namespace.

2.22.1 Top level configuration files

The top level (config) directory typically contains the documents listed in Table 2-6.

Table 2-6 Top level configuration files in ../config/cells subdirectory

File	Purpose	Variable scope (merged/ not merged)
plugin-cfg.xml	Web server plug-in configuration settings	Not merged
plugin-cfg-service.xmi	Configuration of automatic plug-in regeneration settings	Not merged

2.22.2 Cell level configuration files

Under individual cell directories, there is a *cell.xml* file that contains configuration data specific to that cell. There are also several cell level documents that contain configuration data for all managed processes running on that node. The cell level directory typically contains the documents listed in Table 2-7.

Table 2-7 Cell level configuration files

File	Purpose	Variable scope (merged/ not merged)
admin-autz.xml	Administrator access authorization settings	not merged
cell.xml	Configuration data specific to the cell	not merged
filter.policy	Java 2 security policy file - permissions defined here will be filtered out in the node (app.policy) and application (was.policy) level files.	not merged
integral-jms-authorizations.xml	Manage access authorizations to resources owned by the WebSphere Application Server Integral JMS Provider	not merged
multibroker.xml	Data replication service settings	not merged
namestore.xml	Cell persistent portion of the namespace	not merged
naming-autz.xml	Naming authorization settings	not merged
pmirm.xml	PMI request metrics	not merged
resources.xml	Resources available for entire cell, e.g. JMS provider resources, JMS connection factories, JMS Destinations, JDBC providers, etc.	not merged
security.xml	Security configuration for the cell	not merged
variables.xml	Variable substitution values to use forprocesses running on the cell	merged
virtualhosts.xml	virtualhosts used by all servers in the cell	not merged

2.22.3 Node level configuration files

Under each individual node directory, there is a *node.xml* file that contains configuration data specific to that node. There are also several node level documents that contain configuration data for all managed processes running on that node. The node level directory typically contains the documents listed in Table 2-8.

Table 2-8 Node level configuration files

File	Purpose	Variable scope (merged/ not merged)
app.policy	Java 2 security file - access policies for all applications on this node.	not merged
library.policy	Java 2 security file - access policies for all libraries used by this node	not merged
namestore.xml	Node persistent portion of the namespace	not merged
node.xml	Configuration data specific to the node	not merged
resources.xml	Resources available for entire node only	merged
serverindex.xml	Lists the servers associated with the node, the applications deployed to each server and the IP port numbers used by each server's services.	not merged
spi.policy	Java 2 security file - access policies for all service provider interface (SPI) classes on this node	not merged
variables.xml	Variable substitution values to use for processes running on the node	merged

2.22.4 Server level configuration files

Under each individual server directory, there is a *server.xml* file that contains configuration data specific to that server. There are also several server level documents that contain configuration data for the services hosted by the server. The server level directory typically contains the documents listed in Table 2-9.

Table 2-9 Server level configuration files

File	Purpose	Variable scope (merged/ not merged)
namestore-cell.xml	Cell persistent portion of the namespace	not merged
namestore-node.xml	Node persistent portion of the namespace	not merged
resources.xml	Resources available on the server	merged
server.xml	Configuration data specific to the server	not merged
variables.xml	Variable substitution values to use for processes running on the server	merged

2.23 Deleting WebSphere Application Server components

In this topic we discuss the tasks required to delete WebSphere Application Server 5.0 for iSeries components.

If you really want to delete components of a license program, first make a backup of both your data and license program. For details, refer to 10.1, “Backing up WebSphere Application Server” on page 408.

To remove just a portion of the WebSphere Application Server product, such as option 3, from your iSeries server, specify the optional part to delete.

- This example removes the components specific to option 3 (the samples gallery) of the product from your iSeries server.

```
DLTLICPGM LICPGM(5733WS5) OPTION(3)
```

Delete the whole WebSphere Application Server licensed program involves these tasks:

1. Delete the licensed program.
2. Clean up the user data.

2.23.1 Deleting the licensed program

If you need to uninstall WebSphere Application Server, you can remove the entire WebSphere Application Server or just portions of the WebSphere Application Server product. If you installed the WebSphere MQ for iSeries and WebSphere MQ classes for Java and JMS products as part of the WebSphere Application Server installation, then you must uninstall these products if you uninstall WebSphere Application Server. For more information, see “Deleting MQ classes for Java and JMS licensed programs” on page 56.

Delete WebSphere Application Server 5.0 for iSeries product

To remove the entire WebSphere Application Server product, there are certain steps you must perform.

Note: If you have installed both WebSphere Application Server V5.0 and WebSphere Application Server Network Deployment V5.0, this process removes both products.

Perform these steps from an OS/400 command line on the iSeries server on which the product is installed:

1. Ensure that your user profile has sufficient authority. See 2.10.1, “Authority requirement” on page 25 for details.
2. Stop all application server processes by using the stopServer script for each running application server. See 6.5.6, “Stopping an application server” on page 124.
3. Stop the WAS subsystem:

   ```
   ENDSBS SBS(QEJBAS5) OPTION(*IMMED)
   ```

4. When the QEJBAS5 subsystem has ended, use the Delete Licensed Program (DLTLICPGM) command to delete the product:

   ```
   DLTLICPGM LICPGM(5733WS5)
   ```

   The QEJBAS5 library and the `/QIBM/ProdData/WebAS5` directory structure are removed from the system when the product is removed.

Deleting MQ classes for Java and JMS licensed programs

To delete the WebSphere MQ for iSeries and WebSphere MQ classes for Java and JMS products, perform these steps after you have deleted WebSphere Application Server:

1. Execute the WRKMQM command from an OS/400 command-line.
2. To delete a queue manager, select option 4 for each queue manager that was defined for WAS.
3. End the WebSphere MQ for iSeries subsystem:

   ```
   ENDSBS SBS(QMQM)
   ```

4. Delete the WebSphere MQ classes for Java and JMS licensed program:

   ```
   DLTLICPGM LICPGM(5639C34)
   ```

5. Delete the WebSphere MQ licensed program:

   ```
   DLTLICPGM LICPGM(5724B41)
   ```

6. Delete the directory `/QIBM/UserData/mqm` and all of its subdirectories.

2.23.2 Cleaning up the user data

Deleting the WebSphere Application Server from the iSeries server removes all the product libraries and directories. User-defined information is not removed and can be reused if you reinstall the product. You can remove the user data manually. User data consists of:

- All directories and files under the /QIBM/UserData/WebAS5/Base directory. You should manually delete any files or directories you no longer need.
- Any HTTP server directives that were added to an HTTP Server Configuration file to enable WebSphere Application Server for that configuration. You can remove these directives either manually or automatically using the instructions below.

These topics describe how to remove the directives for your HTTP server:

- IBM HTTP Server powered by Apache (V5R1)
- IBM HTTP Server powered by Apache (V5R2)
- Lotus Domino Web Server

IBM HTTP Server powered by Apache (V5R1)

To remove HTTP Server powered by Apache directives, perform these steps:

1. Start your JavaScript-enabled browser.
2. In the URL location or address window, type:

   ```
   http://your.server.name:2001
   ```

   Here, *your.server.name* is the host name of your iSeries server.
3. Press Enter.
4. Enter your iSeries user ID and password. Your iSeries user ID must have sufficient authority. See 2.10.1, “Authority requirement” on page 25 for details. The **AS/400 Tasks** page is displayed.
5. Click **IBM HTTP Server for AS/400**. The **IBM HTTP Server for iSeries** page is displayed.
6. Select **Configuration and Administration**. The HTTP Administration graphical user interface (GUI) interface is displayed.
7. At the top left, select **Configuration**. The main configuration page is displayed.
8. Select your Apache server configuration from the **Server** box.

9. Under **Dynamic Content**, select **WebSphere Application Server**. The **Servlets and JavaServer Pages (JSP)** page is displayed.
10. Select **Disable Servlets and JSPs**.
11. Click **OK**.

IBM HTTP Server (V5R2)

To remove HTTP Server directives, perform these steps:

1. Start your JavaScript-enabled browser.
2. In the URL location or address window, type:

   `http://`*`your.server.name`*`:2001`

   Here, *your.server.name* is the host name of your iSeries server. Press Enter.
3. Enter your iSeries user ID and password. Your iSeries user ID must have sufficient authority. See 2.10.1, “Authority requirement” on page 25 for details. The **AS/400 Tasks** page is displayed.
4. Click **IBM HTTP Server for AS/400**. The **IBM HTTP Server for iSeries** page is displayed.
5. Select the **Manage** tab. In the **Server** drop-down list on the upper right, select the HTTP server instance you want to change.
6. Expand **Server Properties** in the left navigation bar.
7. Scroll down and select **WebSphere Application Server**.
8. Select **Disable Servlets and JSPs**.
9. Click **OK**. This removes the WebSphere directives from the Apache configuration file and returns to the main configuration page.

Lotus Domino Web server

To remove Lotus Domino Web server directives, perform these steps:

1. Remove the DSAPI filter configuration from the Domino server document:
   a. From a Lotus Notes® client connected to the Domino server, edit the Domino document, which is found in the Domino server's Domino Directory (names.nsf).
   b. Within the server document, click the **Internet Protocols** tab and then click the **HTTP** tab.
   c. Remove from the **DSAPI filter file names** field: /QSYS.LIB/QEJBAS5.LIB/LIBDOMINOH.SRVPGM.
   d. Save and exit the Domino Server document.
2. Remove the WebSphereInit directive from the notes.ini file.
   a. Enter the Work with Domino Servers (WRKDOMSVR) command on an OS/400 command line.
   b. For the appropriate Domino server instance, specify option 13 (Edit NOTES.INI) to edit the server's notes.ini file.
   c. Remove the WebSphereInit directive from the notes.ini file.
3. Restart the Lotus Domino Web server.

3

Installation of workstation tools

In this chapter we provide an introduction to the workstation tools that you can optionally install on a workstation. WebSphere Application Server V5.0 for iSeries provides a set of tools to assemble, deploy, and manage the applications.

The workstation components include these tools:

- Administrative scripting tool (wsadmin)
- Application assembly tool (AAT)
- Enterprise bean deployment tool (ejbdeploy)
- ANT utilities (ws_ant)
- Tivoli Performance Viewer (tperfviewer, formerly known as Resource Analyzer)
- Log analyzer (waslogbr)

3.1 Application Assembly Tool (AAT)

The Application Assembly Tool (AAT) is a tool delivered with WebSphere Application Server that helps to package the applications according to the J2EE specification. It lets you create EJB modules, Web modules, RAR modules, and application client modules (see Figure 3-1).

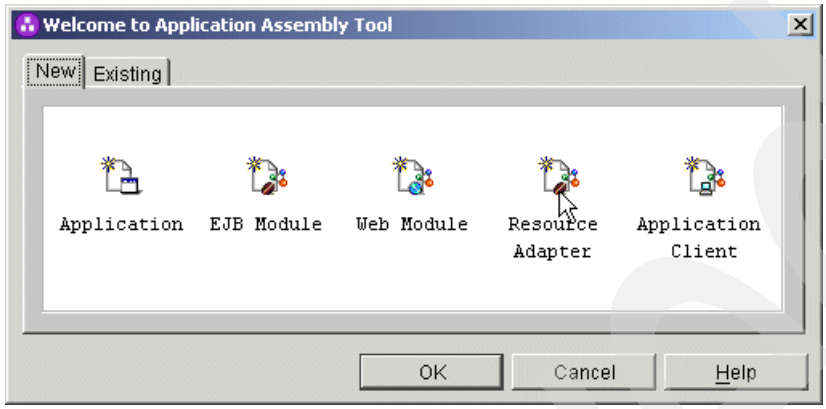


Figure 3-1 Application Assembly Tool

Application assembly is the process of creating an enterprise archive (EAR) file that contains all files related to an application. Application assembly consists of these steps:

- Creating archive files that contain all of the files that belong to a specific application.
- Configuring the runtime behavior of the application.

You can use the AAT to perform these tasks:

- Assemble EJB modules.
- Assemble Web Modules.
- Assemble Application Client Modules.
- Assemble resource adapter modules.
- Assemble or modify modules.
- Migrate application modules from J2EE 1.2 to J2EE 1.3.
- Verify the archive files.
- Generate code for deployment.

An application can consist of a single enterprise archive (EAR) or Web application (WAR) file, or multiple archive files can be bundled into an EAR file if the application has many components. Each bundled entity within the EAR file is called a module. All of the archive files and modules are stored in the standard JAR file format.

These are the types of archives that you can package in an EAR file:

- Enterprise bean (JAR) files (EJB modules)
- Web application (WAR) files (Web modules)
- Application client (JAR) files (client modules)
- Resource adapter (RAR) files (resource adapter modules)

An archive file can contain any of these types of files:

- HTML files
- Servlet class files
- JavaServer Pages (JSP) files
- Enterprise bean class files
- Application client class files
- Image files

Each module also includes an XML file called a deployment descriptor. The deployment descriptor lists the contents and characteristics of the module and contains instructions about how the module is deployed in the runtime environment. To generate the deployment code and validate archive files for enterprise beans, AAT invokes the Enterprise Bean Deployment Tool. For container-managed persistence (CMP) entity beans, the Enterprise Bean Deployment Tool can also generate a map and schema document, and a Data Definition Language (DDL) file containing SQL code for creating a database table.

You can start AAT from the command line of your Windows workstation using the command:

```
assembly.bat
```

3.2 Application Client Resource Configuration Tool

With the Application Client Resource Configuration Tool (ACRCT) you can define references to resources (other than enterprise beans) on the machine where the application client resides.

ACRCT defines resources for the application client. These configurations are stored in the application client EAR file. The application client runtime uses these configurations to resolve and create instances of the resources for the application client. If the client application defines the local resources, run the ACRCT to configure the application accordingly. Use the ACRCT to change the configuration.

You can use the ACRCT to perform these tasks:

- Configure new data source providers.
- Update data source and data source provider configurations.
- Configure mail providers and sessions.
- Update mail session configurations.
- Configure URL providers and sessions.
- Update URLs and URL provider configurations.
- Configure Java messaging client resources.
- Update JMS provider, connection factories, and destination configurations.
- Update MQ JMS provider, MQ connection factories, and MQ destination configurations.
- Configure new environment entries.
- Update Resource Environment Entry and Resource Environment Provider configurations.
- Configure data access for J2EE application clients.
- Remove application client resources.

Note: You cannot use the ACRCT to configure resources for your server.

3.3 Log Analyzer

The Log Analyzer is a GUI tool that permits the user to view any logs generated with log analyzer TraceFormat, such as the IBM service log file and other traces using this format. It can take one or more service logs or trace logs, merge all the data, and display the entries in sequence. More importantly, this tool is shipped with an XML database, the *symptom database*, which contains strings for some common problems, reasons for the errors, and recovery steps (see Figure 3-2). The Log Analyzer compares every error record in the log file to the internal set of known problems in the symptom database and displays all the matches. This allows a user to get error message explanations and information such as why the error occurred and how to recover from it. Refer to 11.3.5, “Monitoring WebSphere Application Server using the Log Analyzer Tool” on page 451 for more information.

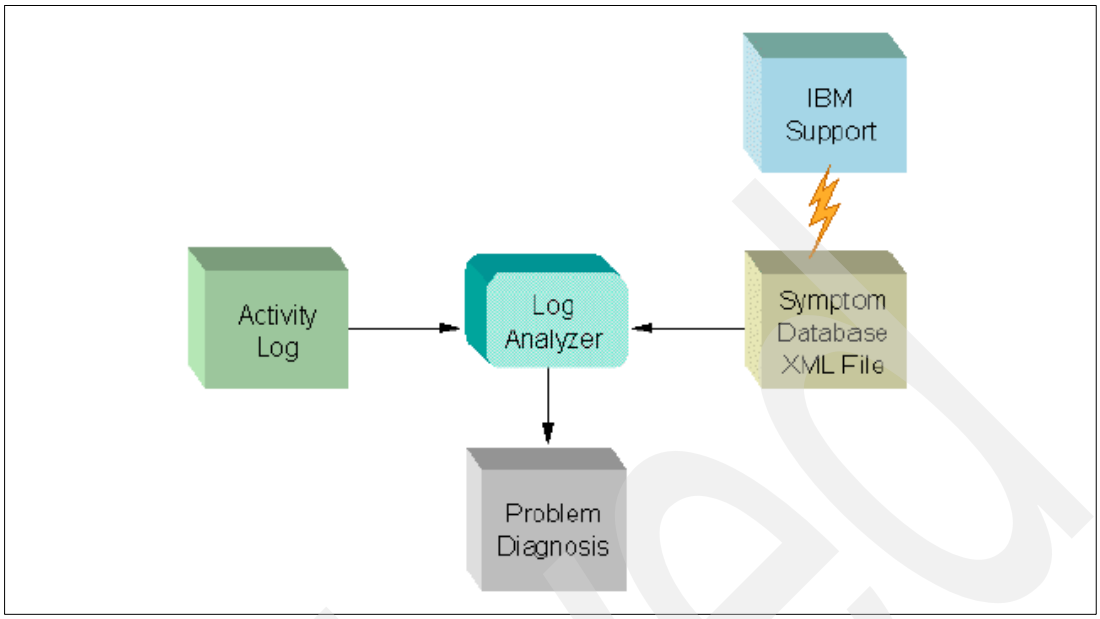


Figure 3-2 Log Analyzer environment

3.4 Tivoli Performance Viewer

Resource Analyzer has been renamed to Tivoli Performance Viewer. Tivoli Performance Viewer is a Graphical User Interface (GUI) performance monitor for WebSphere Application Server. You run Tivoli Performance Viewer on a GUI-capable workstation and connect it to the iSeries application server remotely. Tivoli Performance Viewer is a Java client that retrieves the Performance Monitoring Infrastructure (PMI) data from an application server and displays it in a variety of formats.

The Tivoli Performance Viewer provides access to a wide range of performance data for two kinds of resources:

- Application resources (for example, enterprise beans and servlets)
- WebSphere run-time resources (for example, JVM memory, application server thread pools, and database connection pools)

Performance data includes simple counters, statistical data (such as the response time for each method invocation of an enterprise bean), and load data (such as the average size of a database connection pool during a specified time interval). This data is reported for individual resources and for multiple resources.

Depending on which aspects of performance you are measuring, you can use the Tivoli Performance Viewer to accomplish the following tasks:

- View data in real time or view historical data from log files.
- View data in chart form, allowing comparisons of one or more statistical values for a given resource on the same chart. In addition, different units of measurement can be scaled to enable meaningful graphic displays.
- Record current performance data in a log and replay performance data from previous sessions.
- Compare data for a single resource to a group of resources on a single node.

For more information see WebSphere Application Server InfoCenter at:

http://publib7b.boulder.ibm.com/wasinfo1/en/info/aes/ae/uprf_rtpvgui.html

3.5 Installing the workstation tools

Before you install the WebSphere Application Server workstation components, verify that your hardware and software meets the minimum requirements.

3.5.1 Workstation hardware requirements

If you only plan to use your workstation to administer your WebSphere Application Server environment, you can use any workstation capable of running a Web browser that supports HTML 4.0 and Cascading Style Sheets (CSS).

These workstation hardware requirements apply to the application development and assembly components:

- Capable workstations:
  - An Intel-based personal computer capable of running any of these operating systems:
    - Windows NT Server V4.0 Service Pack 6 (or later)
    - Windows 2000 Server or Advanced Server Service Pack 2 (or later)
    - RedHat Advanced Server for Intel (x86) 2.1
    - SuSE Linux SLES for Intel (x86) V7 Kernel 2.4
    - SuSE Linux for Intel (x86) V7.3 Kernel 2.4
  - A workstation that is capable of running Solaris V8
  - A pSeries™ server that is capable of running AIX® V4.3.3 or V5.1
- Support for a communications adapter or an appropriate network interface
- 120 MB of free disk space (minimum)
- 256 MB of memory (minimum)
- CD-ROM drive

3.5.2 Workstation software requirements

If you only plan to use your workstation to administer the WebSphere Application Server environment, you can use any operating system with a Web browser that supports HTML 4.0 and CSS.

The workstation software requirements listed here apply to the application development and assembly component:

- Any of these operating systems:
  - Windows NT Server V4.0, Service Pack 6 (or higher)
  - Windows 2000 Server or Advanced Server Service Pack 2 (or higher)
  - SuSE Linux for Intel (x86) V7.3 Kernel 2.4
  - SuSE SLES for Intel (x86) V7 Kernel 2.4
  - Sun Solaris V8 (at the latest available maintenance level)
  - AIX V4.3.3 or V5.1
  - Red Hat Advanced Server for Intel (x86) V2.1

- Any of these IBM development kits for Java:
  - Windows NT IBM Enhanced Java Development Kit, Version 1.3
  - HP-UX IBM Software Development Kit for the Java Platform, Version 1.3
  - IBM Developer Kit for Linux, Java 2 Technology Edition, Version 1.3
  - Solaris IBM Java Development Kit, Version 1.3
  - IBM Developer Kit for AIX, Java 2 Technology Edition, Version 1.3

  Note: These IBM development kits for Java are included on the WebSphere Application Server workstation CD-ROM and are installed automatically when you install any of the workstation components of WebSphere Application Server.

- TCP/IP installed and running
- Web browser that supports HTML 4.0 and CSS Cascading Style Sheets (CSS)

3.5.3 Procedure to install workstation tools

To install the workstation components, perform these steps:

1. Place the WebSphere Application Server CD-ROM for your workstation's operating system in your workstation CD-ROM drive. For example, if you are using Windows NT, insert the WebSphere Application Server 5.0 for Windows NT CD-ROM.
2. The LaunchPad wizard should automatically start. If it does not, run the LaunchPad wizard program by navigating to you CD-ROM drive, and invoking the launchpad.bat (Windows) or launchpad.sh (UNIX) program located in the disk1 subdirectory.
3. Select the language version for the LaunchPad wizard and click **OK**.
4. The installation wizard and a Welcome page appears.
5. Click **Install the product**.
6. Select the language version for the installation wizard.
7. Click **OK**.
8. The installation wizard and a Welcome page appears.
9. Click **Next**.
10. Read the license agreement in the Software License Agreement window. To continue, select the *I accept the terms in the license agreement* radio button and click **Next**.
11. The installation wizard will check the prerequisites. If it detects another version of WebSphere Application Server installed on your workstation, a screen comes up listing that version. The wizard also allows to select several more options:
    a. Manually specify the location of another version of WAS
    b. Migrate the previous configuration of WAS to WebSphere Application Server V5
    c. Modify ports for coexistence
12. Ignore these options and click **Next**.
13. On the next screen select **Custom** and click **Next**.
14. On the Choose Application Server components panel, choose the options you want to install. By default, all options are selected. Deselect the options you do not require. On workstations you are using for application development, generally you install the **Application Assembly and Deployment Tools** option. On workstations from which you wish to monitor an application server, generally you install the **Performance and Analysis** options, specifically the **Tivoli Performance Viewer** and **Log Analyzer** options (see Figure 3-3).

Note: Components that are required by the options you select are automatically selected for you. **Deploy Tool** requires WebSphere Application Server installation.

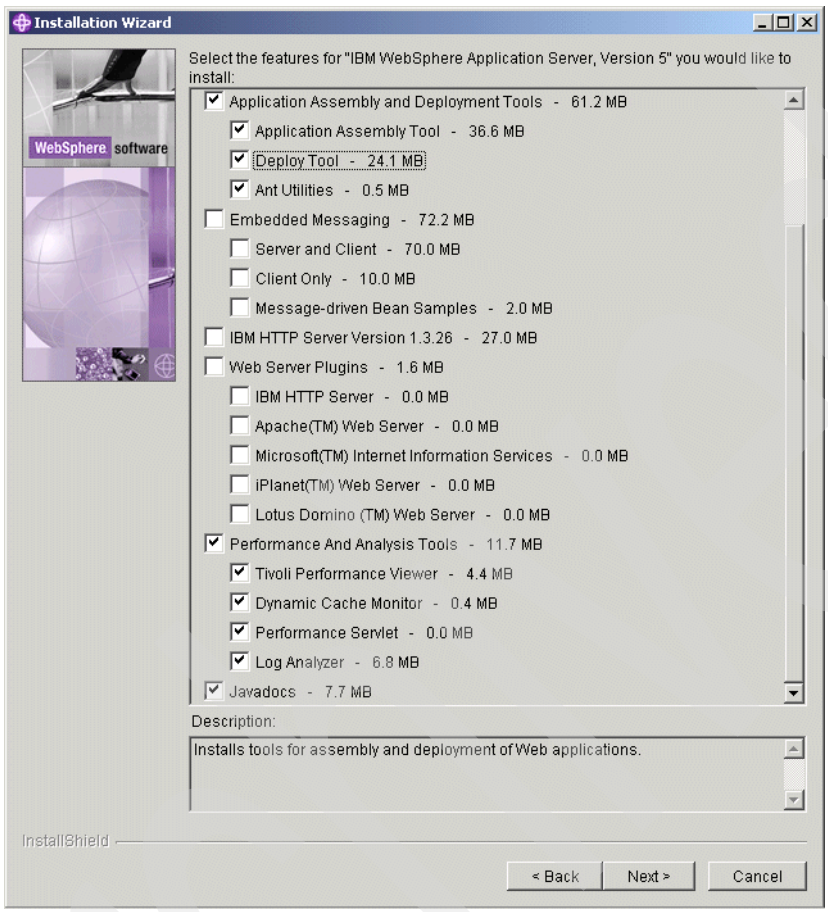


Figure 3-3 Installation Options panel

15. Click **Next**.
16. Type the name of the destination directory and click **Next**. This is the directory in which the workstation components are installed. The field may already be filled in with a default directory name.
17. If the Node Name window is displayed, click **Next** to accept the default node and host names.
18. If the Run application server as a service window is displayed, clear the checkbox, and click **Next**.
19. The summary screen will be displayed. Check the components that will be installed. If you're satisfied, click **Next**. Otherwise, click **Back** and change the installation options.
20. The installation begins. When it completes, the wizard displays a screen with the registration option checked. Click **Next** (you may want to register the product).
21. Click **Finish**.
22. The First Step wizard is displayed. Click **Exit**. This conclude the installation of workstation tools.

4

Installation of WebSphere Application Server Network Deployment V5 for iSeries

In this chapter we show how to install WebSphere Application Server Network Deployment V5.0 for iSeries (WAS-ND) on an iSeries server. We explain the tasks and requirements for installing the WebSphere Application Server Network Deployment, as well as the minimum hardware configuration of your iSeries server that is requested. The prerequisite software licensed program products are also covered.

Additionally, we take you through the planning and installation for WebSphere Application Server 5.0 for iSeries, Network Deployment, in the iSeries environment. Multi-language support, file structure, library usage, and subsystems are also addressed. Since WebSphere 5.0 can be installed in a number of different configurations, we provide a summary of the decisions that must be made when planning an installation to match a particular topology or architecture. This chapter is organized into the following sections:

- Planning your installation
- Performing pre-installation tasks
- Performing WebSphere installation
- Performing Post installation tasks
- Uninstalling WebSphere

4.1 Installation options

Before you install WAS-ND, decide which options you require. By default, all products and all options are installed. You can install a subset of products and options.

WebSphere Application Server Network Deployment V5.0 for iSeries (5733WS5)

WAS-ND product includes the following options:

- **Option Base:** This option is required. It contains the Readme file and other common files for the product.
- **Option 5 (Network Deployment):** This option is required. Option Base must be installed prior to installing option 5. Option 5 contains the Network Deployment runtime.
- **Language option:** The language option denotes the National Language Support (NLS) option for the product.

WebSphere MQ classes for Java and JMS V5.3 for iSeries (5639C34)

This product is not required to run Network Deployment. Install this product if you want to use JMS from your applications or the embedded JMS server; see 2.5, “Considering if you need Java Messaging Service (JMS)” on page 18. The following options are included:

- **Option Base:** Install this option if you want to use JMS from your applications or the embedded JMS server.
- **Option 1 (samples):** Option 1 provides the samples for WebSphere MQ classes for Java and JMS.

Language option

The language option denotes NLS for the product. It is installed only for Option *BASE. English (2924) is the only available language.

Evaluate the available installation options and determine which options you require.

4.2 Prerequisites for WAS Network Deployment

Before you install WebSphere Application Server Network Deployment on your iSeries, verify that your hardware and software meet the minimum requirements.

4.2.1 iSeries and AS/400 hardware requirements

Use the IBM Workload Estimator for iSeries on http://as400service.ibm.com/estimator for help with estimating all system configurations. Systems that do not meet the recommended minimums may be used in environments that support a limited number of users and where longer server initialization times are acceptable.

WebSphere Application Server Network Deployment

These are the recommended minimum server models:

- AS/400e server 170 with processor feature 2292
- AS/400e server 720 with processor feature 2061
- iSeries Model 270 with processor feature 2250
- iSeries Model 820 with processor feature 2395
- 750 MB of memory (recommended minimum)

Note: These requirements are based on a single WebSphere Application Server Network Deployment instance. Additional WebSphere Application Server instances or WebSphere Application Server Network Deployment instances running concurrently will require additional resources.

These requirements represent the recommended minimum requirements. Large deployments or those which must support many users or require shorter response times may need additional resources.

If you are also installing WebSphere Application Server (WAS) on the same iSeries server, refer to 2.2.1, “iSeries and AS/400 hardware requirements” on page 13 for the requirements for the WAS hardware requirements.

The disk requirements for the WAS ND product are given in Table 4-1.

Table 4-1 Disk requirements

WebSphere Application Server for iSeries		
Installation option	**Description**	**After installation**
*BASE	WebSphere Application Server	15 MB
Option 5	Network Deployment	350 MB
WebSphere MQ classes for Java and JMS V5.3 for iSeries		
Installation option	**Description**	**Disk space after installation**
*BASE	WebSphere MQ classes for Java and JMS	15 MB
Option 1	WebSphere MQ classes for Java and JMS - Samples	1 MB

4.2.2 iSeries and AS/400 software requirements

As a basic requirement, OS/400 Version 5 Release 1, or later (in an unrestricted state) is needed to install WebSphere Application Server 5.0 Network Deployment.

Note: For V5Rx OS/400 license programs, we use 5722-xxx as program product numbers.

iSeries and AS/400 required software

The following software is required:

- **OS/400 Version 5 Release 1 (V5R1) or Version 5 Release 2 (V5R2)**:
  The iSeries server must be in an unrestricted state.
- **IBM Developer Kit for Java Version 1.3 (5722-JV1 option 5)**
- **OS/400 Qshell (5722-SS1 option 30)**:
  Required for local installation and to use scripts in Network Deployment.
- **OS/400 Host Servers (5722-SS1 option 12)**:
  Required for remote installation.

iSeries and AS/400 optional software

The following software is optional:

- **OS/400 Digital Certificate Manager (5722-SS1 option 34)**:
  Not required for installation, but required if you plan to use Secure Sockets Layer (SSL) protocol.
- **A Cryptographic Access Provider (5722-AC3):**
  Not needed for installation, but required if you plan to use SSL.
- **DB2(R) Query Manager and SQL Development Kit for iSeries (5722-ST1)**:
  Can be helpful in developing client applications.

4.3 Planning installation for WAS ND on iSeries server

As WebSphere Application Server 5.0 for iSeries, WebSphere Application Server 5.0 Network Deployment for iSeries can be installed in a number of different configurations. This section provides a step-by-step summary of the decisions that need to be made as you plan your installation. Prior to performing an installation, you should consider each of the following options, as the effects of each method and task used during the installation are quite different:

- Do you plan to install WebSphere Application Server 5.0 for iSeries, Network Deployment and WebSphere Application Server 5.0 (Base) for iSeries on the same box?
- Are you migrating an existing installation?
- Should the GUI or a silent installation be used?
- Should a typical or a custom installation be used?
- Is customization of prerequisite checking required?

4.3.1 Planning your steps and time needed for installation

Good planning is half of the success. Before you install WebSphere Application Server Network Deployment V5.0 for iSeries, use Table 4-2 to plan enough time for each step of the process. These times are approximate. Depending on your server, the process may take more or less time.

Table 4-2 WAS-ND installation planning

Task: Planning the installation	Estimated time
Verifying hardware and software prerequisites; see 4.2.1, “iSeries and AS/400 hardware requirements” on page 68 and 4.2.2, “iSeries and AS/400 software requirements” on page 69.	
Obtaining the product and current fixes.	up to 2 weeks
Considering if you need to install Java Messaging Service (JMS); see 4.5, “Considering if you need Java Messaging Service (JMS)” on page 71.	15 minutes
Checking for previous installed WAS versions; see 4.6, “Checking for the previously installed products” on page 71.	5 minutes
Reading Release Notes and migration instructions.	1-2 hours
Evaluating the available installation options and determine which options you require, see 4.1, “Installation options” on page 68.	30 minutes

Installing all prerequisite software; see .4.2.2, “iSeries and AS/400 software requirements” on page 69 Also, install the OS/400 cumulative PTF package; see 2.8, “Installing the correct cumulative PTF package” on page 21.	1-2 hours
Configuring TCP/IP; see 4.8, “Configuring and starting TCP/IP” on page 72.	20 minutes
Installing Network Deployment on your iSeries server. Dependent on the method used. Installing the product from the CD-ROM drive of your iSeries server takes less time than installing it from a remote workstation.	45-120 minutes
Installing WAS ND group PTF.	Up to 2 hours
Doing an IPL of the system if needed.	
Configuring WebSphere Application Server instance.	30-60 minutes
Configuring and starting the Network Deployment domain.	30 minutes
Verifying the installation.	10 minutes

Notes:

- Depending on your server, the process may take more or less time.
- Our testing is done on a 820 machine with V5R2 OS/400 running. The time for installation of WebSphere Application Server Network Deployment V5.0 for iSeries from the iSeries CD-ROM was only about 30 minutes.

4.4 Obtaining WAS Network Deployment and current fixes

For information on how to order WebSphere Application Server Network Deployment for iSeries see 2.4, “Obtaining WebSphere Application Server and current fixes” on page 16.

4.5 Considering if you need Java Messaging Service (JMS)

Decide if you need to install the WebSphere MQ classes for Java and JMS V5.3 for iSeries. For information on when to install the embedded JMS server, refer to 2.5, “Considering if you need Java Messaging Service (JMS)” on page 18.

4.6 Checking for the previously installed products

Refer to 2.6, “Checking for previously installed versions of WAS and MQSeries” on page 19 for more information.

4.7 Reading product Release Notes for important information

Read the product Release Notes for important information about the product. For links to the Release Notes, see the WebSphere Application Server documentation page:

http://www.ibm.com/eserver/iseries/software/websphere/wsappserver/docs/docws50.html

4.8 Configuring and starting TCP/IP

Do the same steps as described in 2.9, “Starting, configuring, and verifying TCP/IP” on page 21 to setup your TCP/IP configuration.

4.9 Installing WebSphere Application Server Network Deployment V5.0 for iSeries

You can use one of these methods to install Network Deployment on your iSeries server:

- From the CD-ROM drive of your iSeries server. Installing the product from the CD-ROM drive of your iSeries server requires direct physical access to the server. The local installation requires less time to complete than a remote installation. You start the local installation with either the Qshell SETUP script or the Run Java (RUNJVA) command.
- From the CD-ROM drive of a workstation with a Microsoft Windows 32-bit operating system. Installing the product from the CD-ROM drive of a workstation does not require direct physical access to the iSeries server. Remote installation requires more time to complete than a local installation.

4.9.1 Authority requirements

To install WAS-ND, you need an iSeries user profile where the user class needs to be set to *SECOFR and the special authorities to *USRCLS.

4.10 Installing WAS-ND from CD-ROM drive of iSeries server

We can perform a local installation of WAS-ND with the SETUP script in Qshell or with the Run Java (RUNJVA) command on an OS/400 command line.

To install WAS-ND from the CD-ROM drive of your iSeries server, follow these steps:

1. Sign on to the iSeries system with a user profile that has sufficient authority. See 4.9.1, “Authority requirements” on page 72 for detail.
2. Run either the script or the command:
   a. The SETUP script in Qshell
      Use this option if you prefer a shorter command.
   b. The Run Java (RUNJVA) command on an OS/400 command line
      Use this command if you prefer control language (CL) commands.

4.10.1 Installing via the SETUP script in Qshell

Perform these steps to install WebSphere Application Server Network Deployment from Qshell using the SETUP script.

1. Place the WebSphere Application Server V5.0, Network Deployment for iSeries CD-ROM in the CD-ROM drive of your iSeries server.
2. Enter `STRQSH` on an OS/400 command line to start the Qshell Interpreter.
3. Enter `cd /QOPT/WEBSPHERE` to change directories to the root installation directory on the CD-ROM.

4. Run the SETUP script to start the installation program. The default is to install all products (WebSphere Application Server Network Deployment for iSeries and WebSphere MQ classes for Java and JMS V5.3 for iSeries) with all supported code options, and the primary language of the iSeries server if supported by the corresponding products or English 2924 if the primary language is not supported by the product. If you do not want to install all of the products, or you want to install a different language, you can specify optional parameters on the command. For a list of the available parameters, see 4.10.3, “Parameters for local installations of WAS-ND” on page 74. To accept the default settings for the installation program, enter this command:

   ```
   SETUP
   ```

   Note: Do not issue any other commands, unless prompted, until the installation is completed. Doing so may cause the installation to stop prematurely.

5. After you enter the SETUP command, messages are displayed that indicate the progress of the installation process. When the setup program completes, press F3 to exit.

Example 4-1 shows a partial sample of output from the installation process.

Example 4-1 Output from the SETUP script

```
> cd /QOPT/WEBSPHERE
$
> SETUP
Loading installation program. Please wait.
Checking current configuration. Please wait.
Installing selected options. Please wait.
Note: This may take up to 120 minutes to complete.
Product 5639C34: WebSphere MQ classes for Java and JMS V5.3 for iSeries
(Code: Option Base)
   Copying stream file to save file.
   Restoring licensed program.
(Language: Option Base)
   Copying stream file to save file.
   Restoring licensed program.
(Code: Option 1)
   Copying stream file to save file.
   Restoring licensed program.
Product 5733WS5: WebSphere Application Server V5.0, Network Deployment for iSeries
(Code: Option Base)
   Copying stream file to save file.
   Restoring licensed program.
(Language: Option Base)
   Copying stream file to save file.
   Restoring licensed program.
(Code: Option 5)
   Copying stream file to save file.
```

4.10.2 Installing via the Run Java (RUNJVA) command

To install Network Deployment to your iSeries using RUNJVA, perform these steps:

1. Place the WebSphere Application Server Network Deployment, V5.0 for iSeries CD-ROM in the CD-ROM drive of your iSeries server.

Notes:

- Do not use the IBM WebSphere Application Server Network Deployment Version 5 For Windows NT and Windows 2000 CD-ROM, the WebSphere Application Server 5.0 for iSeries CD-ROM, or any other CD-ROM (which was shipped with your WebSphere Application Server Network Deployment for iSeries package) for this set of steps.
- Your user profile must have sufficient authority. See 4.9.1, "Authority requirements" on page 72 for detail.

2. Enter the following command to start the installation program. The default is to install all products (WebSphere Application Server Network Deployment and WebSphere MQ classes for Java and JMS V5.3) with all supported code options, and the primary language of the iSeries server if supported by the corresponding products or English 2924 if the primary language is not supported by the product. If you do not want to install all of the products, or you want to install a different language, you can specify optional parameters on the command. For a list of the available parameters, see 4.10.3, "Parameters for local installations of WAS-ND" on page 74. To accept the default settings for the installation program, you would enter this command, using the same capitalization as shown (enter command on a single line):

```
RUNJVA CLASS(SETUP) CLASSPATH('/QIBM/ProdData/OS400/jt400/lib/jt400Native.jar:
/QOPT/WEBSPHERE/OS400:/tmp/WebSphere/WS5INSTALL.JAR') PROP((os400.runtime.exec
QSHELL) (java.version 1.3))
```

Note: If you type an error in this command, such as a wrong classpath, you will get error message JVA0122 like this:

```
Java program completed with exit code 106
```

After you enter the RUNJVA command, messages are displayed that indicate the progress of the installation process. The messages are very similar to the ones shown in Example 4-1.

4.10.3 Parameters for local installations of WAS-ND

The SETUP script in Qshell and the Run Java (RUNJVA) command include several optional parameters that you can use to customize your installation. If a parameter is not specified, the installation program uses the default value for that parameter. If no parameters are specified, WebSphere Application Server Network Deployment for iSeries and WebSphere MQ classes for Java and JMS V5.3 for iSeries are installed with their default options.

For example, you can use parameters to install only the Network Deployment product and skip the installation of WebSphere MQ classes for Java and JMS for iSeries. To do this, you would enter one of these commands to start the installation program:

- From Qshell Interpreter:

```
SETUP -wmqjava -skip true
```

- From an OS/400 command line (enter command on a single line):

```
RUNJVA CLASS(SETUP) PARM('-wmqjava' '-skip' 'true') CLASSPATH
('/QIBM/ProdData/OS400/jt400/lib/jt400Native.jar:/QOPT/WEBSPHERE/OS400:
/tmp/WebSphere/WS5INSTALL.JAR') PROP((os400.runtime.exec QSHELL) (java.version 1.3))
```

Here, the options specified within PARM() are the installation options. The other parameters are required.

Table 4-3 describes all of the configurable parameters for the WebSphere Application Server Network Deployment and the WebSphere MQ classes for Java and JMS V5.3 for iSeries installation. Each product is represented by its product parameter name and its possible parameter values are represented by product assigned parameters. Use the -help parameter to display a list of possible parameters and their usage. The table lists the supported national language options, including double-byte character sets (DBCS) and multi-national character sets (MNCS).

Table 4-3 Installation options

Product parameter name	Product assigned parameters	Possible values	Default	Description
-was	-skip	false true	**false**	If set to true, this product is not installed.
	-language	2980 - Brazilian Portuguese 2950 - English Uppercase only 2924 - English 2938 - English Uppercase only DBCS 2894 - English DBCS 2928 - French 2940 - French MNCS 2929 - German 2939 - German MNCS 2932 - Italian 2942 - Italian MNCS 2962 - Japanese DBCS 2986 - Korean DBCS 2989 - Simplified Chinese DBCS 2931 - Spanish 2987 - Traditional Chinese DBCS	***PRIMARY, if available; otherwise 2924 (English)**	Determines the language to install. If the language parameter is not specified, the installation program detects the primary language on the iSeries server. If that language is supported by the WebSphere Application Server Network Deployment product, the program installs it. If the primary language is not supported, the installation program installs 2924 (English).
	-component	**all** code language	**all**	Selects which components to install. The code component installs only the code. The language component installs only the language options (29nn). The value all includes both the code and language components.
	-option	**all** base 5	**all**	Selects which options to install. The base option installs the common code. Option 5 installs the Network Deployment code. The value all includes the options base and 5. Installation of option 5 requires that the base option is also selected or already installed.

Product parameter name	Product assigned parameters	Possible values	Default	Description
-wmqjava	-skip	**false** true	**false**	If set to true, this product is not installed.
	-component	**all** code language	**all**	Selects which component to install. The code component installs only the code. The language component installs only the language option 2924. The value all includes both the code and language components.
	-option	**all** base 1	**all**	Selects which options to install. The base option installs the server code Option 1 installs the samples code. The value all includes the options base and 1. Installation of option 1 requires that the base option is also selected or already installed.

4.11 Installing WAS ND for iSeries from a workstation

This section describes how to install WebSphere Application Server Network Deployment V5.0 for iSeries on your iSeries server from a Windows 32-bit operating system workstation. There are 2 ways:

1. **Abstract windowing toolkit (AWT):** The AWT mode uses a graphical user interface (GUI) to complete the installation. The AWT mode is the default mode when you install Network Deployment from a workstation.
2. **Silent Install:** The silent mode allows you to enter a single command to start the installation. The silent mode allows you to specify a file which contains the installation option values you want to use. For any option not specified or commented out within the response file, the installer uses the default values.

4.11.1 Installing Network Deployment from a workstation in AWT mode

To use the abstract windowing toolkit (AWT) mode to install Network Deployment from a Windows 32-bit workstation, perform these steps:

1. Verify that the host server jobs have been started on your iSeries server. The host server jobs allow the installation code to run on iSeries.

   Enter this command on an OS/400 command line:

   ```
   STRHOSTSVR SERVER(*ALL)
   ```

2. If TCP/IP is not started, or if you don't know if TCP/IP is started, enter the Start TCP/IP (STRTCP) command on an OS/400 command line.
3. Place the ***WebSphere Application Server Network Deployment, V5.0 for iSeries*** CD-ROM in the CD-ROM drive on the workstation.

   The InstallShield program should automatically start. If it does not, open Windows Explorer and select your CD-ROM drive. Double-click the SETUP.EXE file to start the InstallShield program
4. The Welcome screens is displayed. Read information and click **Next**.

5. Enter the name of the iSeries server where you are installing WAS-ND. You also have to enter a valid user ID and password for the server. This user ID must have sufficient authority (see 4.9.1, “Authority requirements” on page 72 for detail). Click **Next**.
6. On the next panels, select the installation options for WAS-ND.
7. Click **Next**.

> **Note:** If you have WebSphere Application Server V5.0 for iSeries installed on this system, make sure the QEJBAS5 subsystem is shut down.

8. On the next panel, select the installation options for WebSphere MQ classes for Java and JMS V5.3 for iSeries.
9. Click **Next**.
10. Verify the selected options on the next panel (see Figure 4-1). If they are not correct, click **Back** to change your installation options.

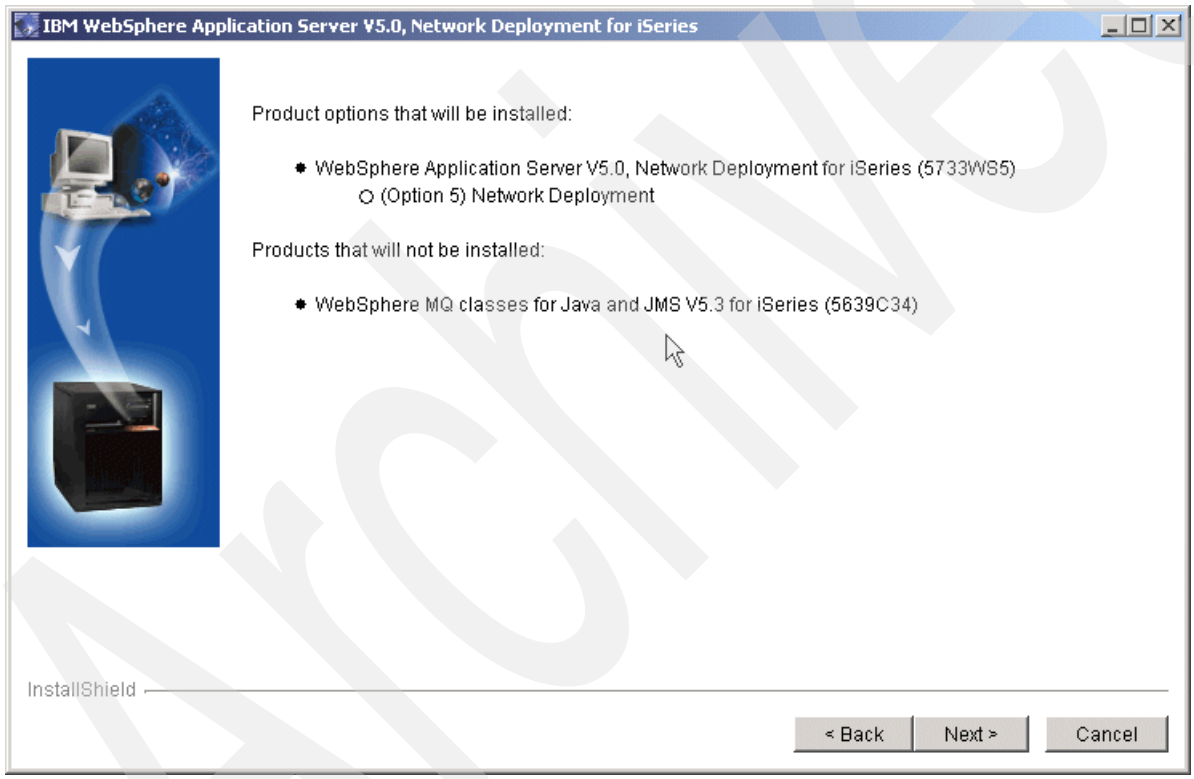


Figure 4-1 Confirm options to be installed

11. If the options are correct, click **Next**.
12. The InstallShield program displays messages that indicate the status of the installation and a status bar to show the progress of the installation.

> **Note:** When you start the installation, a command window is opened in the background. InstallShield logs all messages to this command window. The messages are very similar to the ones shown in Example 4-1 on page 73.

13. After the installation is complete, the summary panel is displayed showing the options that were installed.

14. Click **Finish** to close the InstallShield program.
15. For security purposes, if the host servers were not running, you should return your iSeries server to its original state with the End Host Server (ENDHOSTSVR) command after the install is complete.

4.11.2 Installing WAS Network Deployment from a workstation in silent mode

If you want to install WebSphere Application Server Network Deployment V5.0 for iSeries onto a number of servers with same configuration, silent mode installation is a good choice. The silent mode allows you to install the product with a single command. Non-default installation options must be specified in a response file. During the installation, you won't be able to change the installation options. The process is very similar as for WebSphere Application Server V5.0 for iSeries (see 2.12.2, "Installing WebSphere Application Server using silent mode" on page 35), but the parameters are slightly different.

Parameters for remote silent installations of WAS-ND

Table 4-4 describes all of the possible parameters for the WAS-ND remote silent install response file. The table lists the supported national language options, including double-byte character sets (DBCS) and multi-national character sets (MNCS).

Table 4-4 Installation options for a silent mode

Bean ID	Bean property	Possible values	Default	Description
DestinationBean	system			Determines the destination iSeries server on which the installation is to be performed.
	username			Determines the user name to be used on the destination iSeries server.
	password			Determines the password to be used on the destination iSeries server.

WASSelectBean	skip	**false** true	false	If set to true, WebSphere Application Server Network Deployment is not installed.
	language	2980 - Brazilian Portuguese 2950 - English Uppercase only 2924 - English 2938 - English Uppercase only DBCS 2894 - English DBCS 2928 - French 2940 - French MNCS 2929 - German 2939 - German MNCS 2932 - Italian 2942 - Italian MNCS 2962 - Japanese DBCS 2986 - Korean DBCS 2989 - Simplified Chinese DBCS 2931 - Spanish 2987 - Traditional Chinese DBCS	*PRIMARY, if available; otherwise 2924 (English)	Determines the language to install. If the language parameter is not specified, the installation program detects the primary language on the iSeries server. If that language is supported by the WebSphere Application Server Network Deployment product, the program installs it. If the primary language is not supported, the installation program installs 2924 (English).
	component	**all** code language	all	Selects which components to install. The code component only the product code. The language component installs only the language options (29nn). The value all includes both the code and language components.
	option	**all** base 5	all	Selects which options to install. The base option installs the common code. Option 5 installs Network Deployment code. The value all includes base and Option 5. **Note:** Installation of option 5 requires that the base option is also selected or already installed.

WMQJavaSelect Bean	skip	**false** true	false	If set to true, WebSphere MQ classes for Java and JMS is not installed.
	component	**all** code language	all	Selects which component to install. The code component installs only the product code. The language component installs only the language option 2924. The value all includes both the code and language components. **Note:** The language parameter only applies to the base option of WebSphere MQ class for Java and JMS. The other options are in English only.
	option	**all** base 1	all	Selects which options to install. The base option installs the server code. Option 1 installs the samples code. The value all includes the options base and 1. **Note:** Installation of option 1 requires that the base option is also selected or already installed.

4.12 Installing WAS ND group PTF

After installing the WAS-ND software, install the WAS-ND group PTF in the same way as described in 2.14, “Installing WebSphere group PTF” on page 41.

4.13 Troubleshooting the installation

WebSphere Application Server offers several methods you can use to troubleshoot problems. Which method you use depends on the nature of the problem. Generally, you use a combination of several methods to determine the cause of a problem and then decide on an appropriate method for its resolution. For more information, refer to Chapter 11, “Troubleshooting” on page 431.

4.14 Installed licence programs

If you run the DSPSFWRSC command after installing WAS-ND, you should see the same information as shown in Figure 4-2.

```
                          Display Software Resources
                                                              System:   RCHAS02B
Resource
   ID     Option  Feature  Description
5733WS5   *BASE    5050    WebSphere Application Server V5.0
5733WS5   *BASE    2924    WebSphere Application Server V5.0
5733WS5   1        5050    WAS V5.0 Client development and runtime
5733WS5   1        2924    WAS V5.0 Client development and runtime
5733WS5   2        5050    WAS V5.0 Application server runtime
5733WS5   2        2924    WAS V5.0 Application server runtime
5733WS5   3        5050    WAS V5.0 Samples
5733WS5   3        2924    WAS V5.0 Samples
5733WS5   5        5105    WAS V5.0 Network Deployment
5733WS5   5        2924    WAS V5.0 Network Deployment
```

Figure 4-2 WebSphere application and ND software components

4.15 The Library and Subsystem structure of WAS-ND

This section describes the product library and subsystems that WebSphere Application Server Network Deployment V5.0 for iSeries uses on the iSeries server. Understanding of this structure can help you to facilitate the administration, development, and troubleshooting tasks.

4.15.1 The product library

The iSeries server product library for WAS-ND is the same library as for the WebSphere Application Server 5.0 for iSeries:

```
QEJBAS5
```

4.15.2 Subsystem

WebSphere Application Server 5.0 for iSeries Network Deployment use the following subsystem:

```
QEJBASND5
```

The Deployment Manager process runs in this subsystem.

4.15.3 WAS-ND jobs

This section explains what jobs are started when you work with WAS-ND on iSeries.

Deployment Manager job

The WAS-ND product is shipped configured to use a single deployment manager instance, the `default` instance. The job name for the default Deployment Manager instance is *DMGR*. This job runs in the `QEJBASND5` subsystem.

Multiple instances of WAS-ND are also supported. If you want to create your own WAS-ND instance, the job name for this instance is the first 10 characters of the Deployment Manager's server name.

Note: Remember that the administrative console for WAS-ND is implemented as a Web application. As such, it requires an application server to run.

If the first 10 characters of the application server's name do not provide a valid iSeries job name, the WAS-ND runtime creates a valid job name. If the runtime cannot create a valid job name for the application server, it uses the default job name QEJBSVR.

For more information about the Deployment Manager job, refer to "Verifying that the Deployment Manager has started" on page 275.

JMS Server job

The JMS functions on a node within the WebSphere Application Server administration domain are served by the JMS server on that node. If your application server is part of a cell (Network Deployment domain) managed by the deployment manager and the embedded JMS server is in use, then you should see JMSSERVER job in the QEJBAS5 subsystem.

4.16 IFS directory structure of WAS-ND

WAS-ND uses the integrated file system (IFS) of the iSeries to store product, configuration and application data. The structure and content of the directories are very similar to the WAS directories described in 2.18, "IFS directory structure" on page 46.

For this reason we only mention the differences between two editions:

1. `/QIBM/ProdData/WebAS5/ND`

   Root directory for the WebSphere Application Server Network Deployment product. Files in this directory hold the product data that is shared by all WebSphere Application Server Network Deployment instances. Files under this directory structure should not be modified.

2. `/QIBM/UserData/WebAS5/ND`

   Root directory for WebSphere Application Server Network Deployment instances. All instances of Network Deployment are created under this directory structure.

   Important: The QIBM/ProdData/WebAS5/ND/* is the production directory. Be careful to keep it intact without any modification.

Figure 4-3 shows the configuration repository in the IFS after installation of WAS-ND.

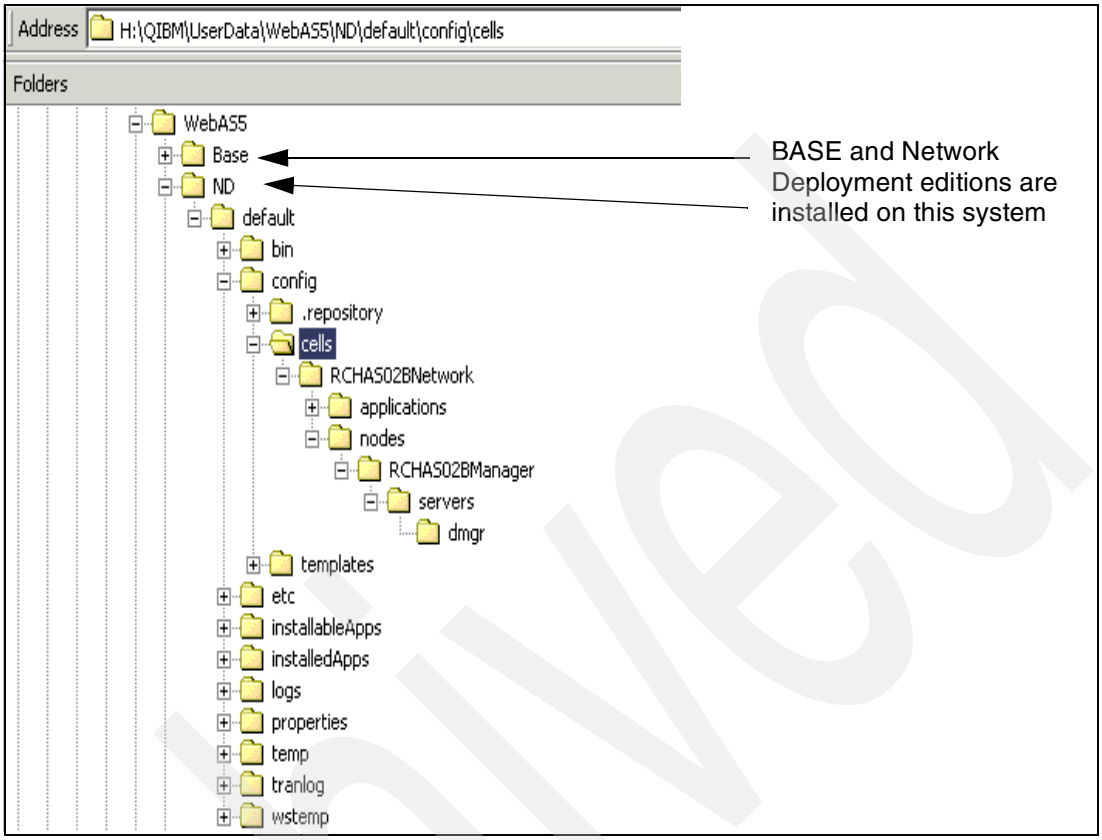


Figure 4-3 IFS ND directory after install

4.16.1 Directory structure of the WAS-ND configuration repository

Deployment Manager (DM) manages all the nodes and application servers that are part of a cell (Network Deployment domain). It is responsible for keeping track of all the configuration changes for the managed application servers and nodes. To achieve this task, DM keeps all the configuration details of a node/application server in its *master repository*. Master repository is a directory structure in IFS under:

```
/QIBM/UserData/WebAS5/ND/instance_name/config/cells
```

In this path, `instance_name` is the name of the DM instance. The structure of the master repository is very similar to the configuration respository of any WAS instance (see 2.19, “WAS instance directory structure” on page 47). When a new node is added to a cell (Network Deployment domain), a new directory structure is added to the master repository of the WAS-ND instance. To keep track of any changes to a node or application server, DM uses a synchronization mechanism. It can be any or both of two methods:

- Automatic synchronization. When enabled, the node agent automatically contacts DM every synchronization interval to attempt to synchronize the node's configuration repository with the master repository owned by the deployment manager.
- Running the *syncNode* script. This allows you to manually synchronize the configuration between a single node and DM for the cell that the node belongs to. The master copy of the configuration documents for the node are copied from the DM’s master repository to the node. Before you run the syncNode script, the node agent and all servers for the node must be stopped.

This script located in /QIBM/ProdData/WebAS5/**Base**/bin.

As you drill down in this directory to the nodes directory, you will see one directory for each node in the cell. Each node will then have a specific subdirectory for each application server. So you can see the structure of the domain reflected in the file system.

Figure 4-4 shows the configuration repository of the WAS-ND instance after adding 2 nodes:

- A node with the *default* instance on RCHAS02B (with the server - server1)
- The *webface* node on RCHAS02B

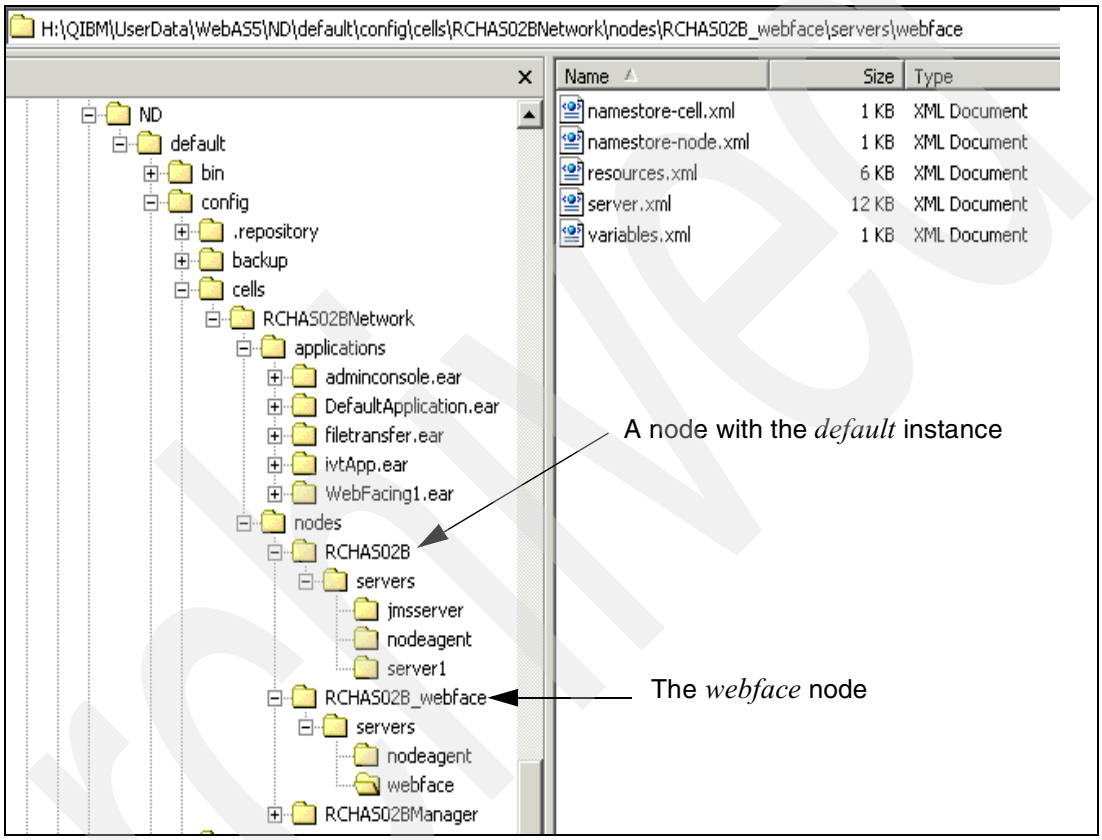


Figure 4-4 IFS ND directory after adding nodes to the cell

4.16.2 Hierarchy of configuration directories

The administrative repository for the Network Deployment instance starts with the config directory located directly under the root directory for the instance. For the default instance, the administrative respository is contained in the /QIBM/UserData/WebAS5/ND/default/config directory.

The configuration files are arranged in a set of cascading directories under this directory. The cell repository hierarchy consists of four tiers of directories. Each of these directories contains several documents relating to different parts of the system.

The /config directory is the root of the configuration repository, containing the directories and files described in the following sections.

/config

The config directory is the repository root directory for a WAS-ND instance. It contains a single file, plugin-cfg-service.xmi. This file defines the custom service that causes the Web server plugin file, plugin-cfg.xml, to be regenerated each time the instance is started.

It also contains the following subdirectories:

- **.repository:** This is for the repository.
- **backup:** When you perform administrative tasks, WAS generates temporary configuration files and backup configuration files in this directory.
- **cells:** Configuration data for the entire cell goes in this directory.
- **temp:** Temporary files are stored in this directory. All uncommitted (unsaved) configuration changes are stored in this directory.
- **templates:** The templates directory contains two subdirectories, named default and system, which contain template XML files for several configuration object types such as servers, JDBC providers and JMS providers.

  The administrative console uses these templates when displaying default properties for a resource you are creating. You can also use these templates with the wsadmin scripting tool to create new resources based on the templates.

/config/cells

The *cells* directory contains a single subdirectory for the cell to which the instance belongs. This directory is named as your iSeries server TCP/IP name following by the name Network. We refer to this name as cellname. It is RCHAS02BNetwork in our example as shown in Figure 4-3 on page 83. The cells directory also contains the WebSphere plugin file, `plugin-cfg.xml`. The WebSphere plugin uses this file to determine which Web resources are installed in your cell.

4.16.3 Variable-scoped documents

Identically named documents that exist at differing levels of the configuration hierarchy are called “variable scoped” documents. In both WebSphere environments, WAS and WAS-ND, the hierarchy, scope and behavior is the same. Refer to 2.22, “Variable-scoped documents” on page 52 for more information.

4.17 Uninstalling Network Deployment

This topic discusses the tasks required to uninstall WAS-ND. If you really want to uninstall the license program, first make a backup on both your date and license program (see 10.1, “Backing up WebSphere Application Server” on page 408).

Uninstalling WAS-ND involves these tasks:

1. Deleting the licensed program
2. Cleaning up the user subdirectories

4.17.1 Deleting the licensed program

If you need to uninstall WebSphere Application Server Network Deployment V5.0 for iSeries, you can remove either the entire WebSphere Application Server V5 product (5733WS5) or just the Network Deployment (5733WS5 option5). If you installed the WebSphere MQ classes for Java and JMS product as part of WAS-ND, then you must uninstall this product if you uninstall Network Deployment (unless you have also installed WebSphere Application Server and are only uninstalling Network Deployment). For more information, see 4.17.3, "Deleting WebSphere MQ classes for Java and JMS licensed program" on page 86.

To remove the entire WebSphere Application Server product, perform these steps from an OS/400 command line on the iSeries server on which the product is installed:

> **Note:** If you have installed both WebSphere Application Server V5.0 for iSeries and WebSphere Application Server Network Deployment V5.0 for iSeries, this process removes both products.

1. Ensure that your user profile has sufficient authority. See 4.9.1, "Authority requirements" on page 72 for detail.
2. Stop the WebSphere Application Server environment.

   ```
   ENDSBS SBS(QEJBAS5) OPTION(*IMMED)
   ENDSBS SBS(QEJBASND5) OPTION(*IMMED)
   ```

3. To delete all options:

   ```
   DLTLICPGM LICPGM(5733WS5)
   ```

   The QEJBAS5 library and the /QIBM/ProdData/WebAS5 directory structure are removed from the system.
4. To remove just the WAS V5 Network Deployment product, type this command on an OS/400 command line and press Enter:

   ```
   DLTLICPGM LICPGM(5733WS5) OPTION(5)
   ```

4.17.2 Cleaning up the user subdirectories

Uninstalling WebSphere Application Server Network Deployment V5.0 for iSeries from the iSeries server removes all the product libraries and directories. User-defined information is not removed and can be reused if you reinstall the product. You can remove the user data manually. User data consists of all subdirectories and files under the directory /QIBM/UserData/WebAS5/ND. You should manually delete any files or directories you no longer need.

4.17.3 Deleting WebSphere MQ classes for Java and JMS licensed program

The WebSphere MQ classes for Java and JMS product, which is shipped with WAS-ND, have a restricted license: This product can only be used with WAS-ND or WAS. If you completely uninstall the WebSphere Application Server product, you must uninstall the WebSphere MQ classes for Java and JMS product. If you have purchased a full license for this product, it does not have to be uninstalled.

To uninstall the WebSphere MQ classes for Java and JMS product, type this command on an OS/400 command line, after you have uninstalled WAS-ND, and press Enter:

```
DLTLICPGM LICPGM(5639C34)
```

5

Configuring the iSeries for WebSphere Application Server 5.0

At this point, you have installed WebSphere Application Server V5.0 for iSeries.

In this chapter, we explain how you can prepare the iSeries to be ready for the initial configurations to be described in Chapter 6, “WebSphere Application Server 5.0: configuration and administration” on page 107 and Chapter 7, “WebSphere Application Server Network Deployment 5.0: configuration and administration” on page 263.

We also outline the OS/400 objects created and used by WebSphere, and discuss some helpful tips on tuning the system environment for the expected workload.

5.1 Preparing for WebSphere Application Server installation

In preparation for the configuration of WebSphere Application Server, you must configure the software license (see 6.4, “Configuring software license information” on page 111), a TCP/IP network, an HTTP server, and so on. These steps are listed in the installation and configuration chapters for the product edition you are using. This section reviews some optional configuration steps to help your system handle the new application server workload more efficiently.

5.1.1 Configuring MAXJOBS for database servers

If your application uses JDBC to access the OS/400 database from your applications, you may need to change the maximum number of jobs allowed for the database servers. Each JDBC connection object requires a server job. Generally, it is easiest to account for the number of jobs you need by setting the maximum number of jobs to *NOMAX. There are two JDBC drivers to access a database on iSeries, and each has its own server jobs. Use the Change Prestart Job Entry (CHGPJE) commands shown below to change the prestart job entry for the database server jobs you intend to use from your application.

IBM Developer Kit for Java (Native JDBC driver)

Used when connecting to the local database of the system running the JVM.

```
CHGPJE SBSD(QSYSWRK) PGM(QSQSRVR) MAXJOBS(*NOMAX)
```

IBM Toolbox for Java (5722-JC1) includes a JDBC driver

Used when connecting to iSeries databases across a network.

Host server with no SSL:

```
CHGPJE SBSD(QUSRWRK) PGM(QZDASOINIT) MAXJOBS(*NOMAX)
```

Host server *with* SSL:

```
CHGPJE SBSD(QUSRWRK) PGM(QZDASSINIT) MAXJOBS(*NOMAX)
```

Note: If you prefer to establish a limit on the number of database connections via this host server, you may instead specify an integer value corresponding to the maximum number of concurrent JDBC connections for the MAXJOBS parameter. Keep in mind that other products and users may require some connections as well.

Tip: Use the Display Active Prestart Jobs (DSPACTPJ) command to see the peak number of concurrent connections for each host server. This may help you establish a reasonable maximum for your system environment.

A message similar to the following is displayed when the maximum number of jobs is successfully changed:

```
Program QSQSRVR found in library QSYS.
Active subsystem description QSYSWRK in QSYS changed.
```

5.1.2 Other CHGPJE parameters for database servers

If you are using the connection pooling features of WebSphere Application Server, then the connection pool manager controls the number of database connections. From the iSeries perspective, there are active connections for each handle available in WebSphere’s connection pool. So the prestart jobs on the iSeries should be configured with that in mind.

These are the parameters that can affect the performance of WAS:

- **STRJOBS**: Start jobs specifies whether the jobs should be started when the subsystem is started. To avoid the need for performing a STRPJ command after each system IPL, we suggest you specify *YES.
- **INLJOBS**: Initial number of jobs specifies the number of jobs that are prestarted when the host server is started. Prestarting them avoids using the resources to start more jobs during peak usage.

  We suggest you specify a value that approximates the expected maximum number of database connections (see Figure 5-1, "DSPACTPJ of the native JDBC driver - screen 1" on page 91, peak number of jobs in use 3).

  If you are using the connection pooling features of the WebSphere Application Server, then the initial number of jobs should be at least as large as the expected number of connections coming in from the connection pool. We suggest setting the initial number of jobs to be 20% larger than the initial size configured for the connection pool. And then we suggest you periodically monitor the average number of prestart jobs in use, as described in 5.1.3, "Reviewing active prestart jobs" on page 90. And set the initial number of jobs to be 20% larger than the average you observe.

  Note: Upon startup, the connection pool manager connects to the database server, then, as connections are requested from it, the pool manager adds connections until it reaches the *minimum* number of connections configured for the pool. So, the connection pool manager doesn't allocate the minimum pool size immediately. But, once it opens that number of connections, it will leave them open.

- **THRESHOLD**: Threshold is the point when you want the system to begin setting up additional prestarted jobs; essentially the lead time before completely running out of prestarted jobs. When the threshold is reached, additional jobs are prestarted (based on the setting of ADLJOBS). The idea is to have the prestarted jobs available before connections come in that require them.

  We suggest that you specify a value that approximates the expected number of new connections during a 60 second period of peak activity. Review the Average wait time (5) as shown in Figure 5-2, and increase the threshold until the wait times are minimal or zero. The existence of prestarted jobs on the system will not degrade performance of other work, so it is better to error on the side of having prestarted jobs available when clients need them.
- **ADLJOBS**: Additional number of jobs. Specifies the number of jobs that will be prestarted when the number of available prestarted jobs falls below the threshold. When using a connection pool, the pool manager reassigns the same physical database connections repeatedly. So, while the application may use many connections from the pool, the pool manager will only be needing additional connections to the database if the other handles in the pool are already in use.

  We suggest you specify a value that is 5 to 10% of your initial number of jobs.

  Tip: Avoid setting this value over 100. The user that connects to the job triggering the threshold will inherit the burden of prestarting all of these additional jobs. Keeping this value below 100 will assure more consistent connection times for the applications.

- **MAXUSE**: Maximum number of uses. Specifies the number of times a given host server job may be used for incoming connections before it is discarded, and a new prestart job is started. The less jobs the system must start, the less overhead is incurred. So reusing the jobs can reduce the overhead of job creation.

  Most environments use the connection pool manager built into WebSphere Application Server. The pool manager remains connected to the database, even as the applications open and close connections from the pool. So, to the iSeries database, the connection is established once, and remains connected for a long period of time. The result is a single connection appears to drive high numbers of unrelated transactions.

  We suggest you establish an expiration timer on the connection pool, so that a given connection handle is only used for a limited time. We also suggest setting the maximum uses down to a level that the database host server job gets discarded after about an hour of use. This releases all of the resources the job has accumulated while processing its various requests.

  For example, set the connection pool manager to expire each connection handle after 20 minutes, and set the MAXUSE to 3. If you are not using connection pooling, we suggest you retain the shipped value of 200 uses.

  If you encounter any problems with side effects resulting from the activities of the prior user of the database connection, you may set this value to 1 to avoid such problems while locating a fix. Keep in mind that this typically increases your requirements for server jobs and the overhead of creating them; especially if you are not using connection pooling. If MAXUSE is set to 1, we suggest you also increase the values for THRESHOLD and ADLJOBS, and work to locate a fix for the reuse problem as soon as possible.

5.1.3 Reviewing active prestart jobs

Once your initial prestart job parameters are set, you can monitor the actual activity to see if any adjustments are needed. To help you get an idea of the actual usage of prestart jobs on your system, and whether you should adjust any of the above parameters, use the Display Active Prestart Jobs (DSPACTPJ) command for the specific host server you wish to study. Use the following commands to display activity of the JDBC host server prestart jobs:

- Toolbox JDBC:

  ```
  DSPACTPJ SBS(QUSRWRK) PGM(QSYS/QZDASOINIT)
  ```

- Toolbox JDBC with SSL:

  ```
  DSPACTPJ SBS(QUSRWRK) PGM(QSYS/QZDASSINIT)
  ```

- Native JDBC:

  ```
  DSPACTPJ SBS(QSYSWRK) PGM(QSYS/QSQSRVR)
  ```

You will see the display shown in Figure 5-1.

```
                        Display Active Prestart Jobs                AS07
                                                          12/01/02  13:16:37
Subsystem  . . . . . :   QSYSWRK         Reset date . . . . . :   11/20/02
Program  . . . . . :   QSQSRVR         Reset time . . . . . :   07:59:32
  Library  . . . . . :     QSYS          Elapsed time . . . . :   0269:17:04

Prestart jobs:
  Current number . . . . . . . . . . . . . . . . . :   29
  Average number . . . . . . . . . . . . . . . . . :   28.6  1
  Peak number  . . . . . . . . . . . . . . . . . . :   29

Prestart jobs in use:
  Current number . . . . . . . . . . . . . . . . . :   25
  Average number . . . . . . . . . . . . . . . . . :   25.0  2
  Peak number  . . . . . . . . . . . . . . . . . . :   26    3




                                                                      More...
Press Enter to continue.

F3=Exit   F5=Refresh   F12=Cancel   F13=Reset statistics
```

Figure 5-1 DSPACTPJ of the native JDBC driver - screen 1

Page down to see the display shown in Figure 5-2:

```
                        Display Active Prestart Jobs                AS07
                                                          12/01/02  13:16:37
Subsystem  . . . . . :   QSYSWRK         Reset date . . . . . :   11/20/02
Program  . . . . . :   QSQSRVR         Reset time . . . . . :   07:59:32
  Library  . . . . . :     QSYS          Elapsed time . . . . :   0269:17:04

Program start requests:
  Current number waiting . . . . . . . . . . . . . :   0
  Average number waiting . . . . . . . . . . . . . :   .0  4
  Peak number waiting  . . . . . . . . . . . . . . :   0
  Average wait time  . . . . . . . . . . . . . . . :   00:00:00.0  5
  Number accepted  . . . . . . . . . . . . . . . . :   28
  Number rejected  . . . . . . . . . . . . . . . . :   0









                                                                      Bottom
Press Enter to continue.

F3=Exit   F5=Refresh   F12=Cancel   F13=Reset statistics
```

Figure 5-2 DSPACTPJ of the native JDBC driver - screen 2

As you can see on the sample displays, the average number of prestart jobs 1 is 28.6, and the average number of jobs in use 2 is 25. So, on average, there were more than 3 prestarted jobs that were available to receive a connection. Unless the system is approaching its limits to the amount of work it can perform, you should try to set the prestart job parameters to keep the Average number waiting 4 at or near zero, as shown in Figure 5-2. In other words, set your threshold and additional number of jobs so that connections don't have to wait for prestart jobs to be created.

5.1.4 Configuring TCP/IP

To configure and run WebSphere Application Server on the iSeries, TCP/IP must be configured properly and must be started before you start the WebSphere Application Server environment. In this section we assume you already have a basic TCP/IP configuration, and review steps to improve its performance. If you need to create a TCP/IP configuration, refer to Chapter 2, "Installation of WebSphere Application Server 5.0 for iSeries" on page 11.

TCP/IP MTU configuration

It is recommended that your line description, TCP/IP interface, and TCP/IP routes all use the same maximum frame size. The following steps are recommended to configure the interface and route to use the maximum frame size from the line description. By following this approach, if you determine a change is necessary, due to network quality or router limitations, you only have to change the value in one place (the line description) and recommendation that they all match is still being followed.

Configuring TCP/IP interfaces to use MTU from line description:

Follow these steps:

1. Enter the Configure TCP/IP (CFGTCP) command on an OS/400 command line, the Configure TCP/IP menu is displayed.
2. Use option 1 to Work with TCP/IP Interfaces.
3. Take option 5 (Display) on the IP address used for your Web serving.
4. Check the maximum transmission unit (MTU). We recommend that you retain the default value of *LIND. If you need to change the value:
   a. Press F12 to return to the list of interfaces.
   b. Use option 2 (Change).
   c. Modify MTU.
   d. Press Enter.
5. Press F12 to return to the Work with TCP/IP Interfaces display.
6. Repeat from step 3 for each IP address used for Web serving.
7. Press F12 again to return to the Configure TCP/IP menu.

Configuring TCP/IP routes to use MTU from TCP/IP interface:

Follow these steps:

1. If you are not already on the Configure TCP/IP menu from the above steps, then enter the Configure TCP/IP (CFGTCP) command on an OS/400 command line, the Configure TCP/IP menu is displayed.
2. Use option 2 to Work with TCP/IP routes.
3. Take option 5 (Display) to display the routes that are used by your Web server.

4. Check the maximum transmission unit (MTU). We recommend that you retain the default value of *IFC to use the MTU from the interface that is associated with this route. If you need to change the value:
   a. Press F12 to return to the list of routes.
   b. Select option *2* (Change).
   c. Modify MTU.
   d. Press Enter.
5. Press F12 to return to the Work with TCP/IP routes display.
6. Repeat from step 3 for each route used by your Web server.
7. Press F12 again to return to the Configure TCP/IP menu.

TCP/IP buffer sizes

The TCP/IP buffer size defines how much data can queue up waiting for an application to receive (or send) it before any further data from the remote system is refused. The process of refusing further data is implemented at the TCP/IP protocol level and is called *flow control*. Flow control can use significant CPU time, and results in additional network latency to wait for the signal to resume transmission. We recommend that the buffer sizes be increased from their default 8KB values to avoid flow control under normal operating conditions. A larger buffer size reduces the potential for flow control to occur, thus improving efficiency of the CPU.

Larger buffer sizes also increase the amount of work that an erroneous client or a malicious attacker can queue up. So, the buffer size can be used as a simple means of limiting the resources that such conditions could consume — the idea being to avoid flow control, but not allow more work to buffer up than the system has the capacity to process.

The buffer size recommended to provide the best throughput depends upon several network environment factors, including types of switches and systems, acknowledgement timing, error rates, and network topology. When transferring large amounts of data, you may experience higher throughput by setting the buffer sizes up to 8 MB (which is the maximum). This will let the data keep flowing, and allow the receiver to accept data in large segments.

It is important to note that using the largest buffer size is *not* appropriate in most environments. If you are only expecting to serve requests and responses of modest size, then a larger buffer size will not improve your throughput. Also, the protocol for most socket servers is to process one request at a time for a given connection. Under such a protocol, the client, if properly following the protocol, won't be queueing up more than one request. Therefore, buffers that exceed the size of a single, large request are not required.

To review or revise the default TCP/IP buffer sizes, enter the Change TCP/IP Attributes (CHGTCPA) command at a command line, and press F4 (Prompt). The buffer sizes are shown on the display as the TCP receive buffer size and the TCP send buffer size.

As a general guideline, we suggest setting both the send and receive buffer sizes to 1 MB. To do this, enter the following command:

```
CHGTCPA TCPRCVBUF(1048576) TCPSNDBUF(1048576)
```

Note: These buffer sizes are used as the default for each TCP/IP conversation that is established. Applications may explicitly control the buffer sizes used for their specific connections by using the SO_SNDBUF and SO_RCVBUF socket options on the setsockopt()--Set Socket Options API.

If you are using the Toolbox JDBC driver, you can configure the buffer sizes you would like to use in the custom properties configured for the data source.

The connection between the HTTP plug-in and the application server is overridden to be 64K, so the buffer sizes specified in the CHGTCPA command above will not affect this particular conversation.

5.2 Configuring your OS/400 environment for performance

In general, a lot of the same performance issues, practices, and guidelines used in configuring systems that run traditional iSeries applications will apply to tuning an iSeries for advanced applications as well. This section focuses on discussing the issues that more directly pertain to Java, JDBC, EJBs, and TCP/IP configurations. For more background information on system tuning, refer to the Systems Management area of the iSeries Information Center.

5.2.1 Pertinent system values

System values are to iSeries jobs what properties are to Java. They define the basic characteristics of the system environment and of all the jobs running on the system.

There are a number of system values you should consider changing with either the Change System Value (CHGSYSVAL) or Work with System Value (WRKSYSVAL) commands:

- The *performance adjustment system value* defines if you are manually tuning the system, or having the system make its own adjustments.
- The *allocation system values* define how the system handles internal structures used as jobs begin and end in the system. Adding the WebSphere Application Server to a system typically does not dramatically change the number of jobs processed in the system, and therefore adjustments to these allocation system values typically do not have a major impact. They are discussed here to present the complete picture of the controls available to you.
- The *storage system values* are often key to resolving bottlenecks or problems in a WebSphere Application Server configuration. This is because the defaults for these values are typically not in-line with the values recommended for an application server installation.

The following sections describe these system values, the recommended settings, and how to determine appropriate values for your own production environment.

Performance adjustment system value

The system has an automated performance adjuster. Its purpose is to reallocate system resources in a manner that provides the greatest overall system throughput. These system resources are main storage, and activity levels available in the system main storage pools. The QPFRADJ system value defines whether the automated performance adjuster is activated or not.

QPFRADJ

This system value may be set to one of the following:

```
0 = No adjustment
1 = Adjustment at IPL
2 = Adjustment at IPL and automatic adjustment
3 = Automatic adjustment
```

The shipped value is 2. We suggest you either use 0 and perform your performance adjustments manually, or 3. For more information on manual performance tuning, see 5.3, "Configuring main storage pools" on page 99. We suggest *avoiding* the values of 1 and 2 for WebSphere application server environments. This is because the values that are computed at IPL may not be adequate to start your HTTP server and application servers, and these automatic adjustments may not step up the activity levels quickly enough. A change to this system value takes effect immediately.

Allocation system values

In this section we discuss how the system controls all the active jobs, how to determine the number of jobs that exist on your system, and then how to tune your allocation system values for your environment.

Every job on the iSeries server has a Work Control Block Table (WCBT) entry that tracks the jobs in the system. A WCBT entry can be seen as a pre-fabricated frame for a job that is filled with the jobs' run attributes as the job becomes active within a subsystem. A WCBT entry is required as long as a job is active, on a job queue, or has any spooled output files.

To review all of the allocation system values, use the following command:

```
WRKSYSVAL SYSVAL(*ALC)
```

In reading the documentation and the help text on these system values, pay particular attention to the phrases *jobs in system* and *active jobs*. **Active jobs** is a reference to jobs that are actively running. **Jobs in system** is a reference to active jobs, plus jobs on job queues, and also jobs that still have any spooled output files remaining on the system.

Internal structures required for a job

There are two internal structures that a job requires to run:

1. A WCBT entry, which is a permanent job structure:
   - Created as soon as a job enters a job queue
   - Exists through the job becoming active
   - Remains until the job's last spooled output file is deleted
2. A temporary job structure:
   - Required only after the job is serviced by the job queue, and goes active
   - Free for reuse once the job ends
   - Requires approximately 110K of storage

Use the Display Job Tables (DSPJOBTBL) command to get an overview of your present system configuration, and present usage of these job structures (see Figure 5-3).

```
                            Display Job Tables                      AS07
                                                          11/29/02  14:49:16
Permanent job structures:                 Temporary job structures:
  Initial  . . . . :   30  1                Initial  . . . . :   20  4
  Additional . . . :   10  2                Additional . . . :   10  5
  Available  . . . :   911                  Available  . . . :   175
  Total  . . . . . :   2080
  Maximum  . . . . :   163520  3


                       ---------------------Entries----------------------
    Table         Size        Total    Available       In-use        Other
        1      2130688         2080          911         1169  6         0









                                                                      Bottom
Press Enter to continue.

F3=Exit   F5=Refresh   F11=In-use entries   F12=Cancel
```

Figure 5-3 DSPJOBTBL initial display

Press F11 to display the in-use entries (see Figure 5-4).

```
                            Display Job Tables                      AS07
                                                          11/29/02  14:49:16
Permanent job structures:                 Temporary job structures:
  Initial  . . . . :   30                   Initial  . . . . :   20
  Additional . . . :   10                   Additional . . . :   10
  Available  . . . :   911                  Available  . . . :   175
  Total  . . . . . :   2080
  Maximum  . . . . :   163520


           -----------In-use Entries------------
                              Job        Output
    Table       Active        Queue       Queue
        1          243  7         0         926









                                                                      Bottom
Press Enter to continue.

F3=Exit   F5=Refresh   F11=Total entries   F12=Cancel
```

Figure 5-4 DSPJOBTBL display with In-use entries

QTOTJOB: Initial total number of jobs

This value should be set high enough not to be exceeded between IPLs. If this value is too low, your system may suffer apparent hang conditions for intermittent short periods. This is due to the suspension of all active jobs while QADLTOTJ (2) additional WCBTs are created (see Figure 5-3 on page 96). Since this impact on the end-users is undesirable, it is generally better to set this value beyond the level that you are likely to attain on your system. To determine a value for QTOTJOB, compare the existing value (1) with the number of In-use entries (6) shown on the DSPJOBTBL display, as shown in Figure 5-3 on page 96, on a day of peak system usage.

The shipped value is 30. This is too small for most customer environments. We suggest you add at least 20% to the peak In-use entries (6) observed on the DSPJOBTBL display.

Attention: Changes to the QTOTJOB system value do not take affect until the next IPL.

QADLTOTJ: Additional number of total jobs

This value specifies how many WCBT entries are created when the number of jobs in the system exceeds the number specified for QTOTJOB. A large number of additional total jobs takes longer to create the additional WCBT entries. Since the active jobs on the system are suspended while the additional entries are created, making this value too large will result in the entire system appearing to be suspended for a long period of time. Making the value too small will result in such suspensions occurring frequently. The present setting is shown on the DSPJOBTBL display (2), see Figure 5-3 on page 96. WebSphere Application Server workloads tend to use large numbers of *threads* rather than *jobs*, so it typically does not impact the need for additional total jobs.

The shipped value is 10. We suggest you retain this default. A change to this system value takes effect immediately.

QMAXJOB: Maximum number of jobs

This value sets an upper limit to the number of jobs in the system. Once this maximum is reached, the QADLTOTJ system value is no longer used to extend the number of jobs the system supports. At that point jobs may not be submitted nor started until In-use Entries (6) as shown in Figure 5-4 on page 96 become available either by jobs ending, or spooled output files being deleted for jobs that have previously ended. The present setting is shown on the DSPJOBTBL display (3), see Figure 5-3 on page 96.

The shipped value is 163520. A change to this system value takes effect immediately. Since no resources are consumed until jobs are actually initiated, and being unable to start new jobs could be detrimental to a production environment, we suggest either retaining the default setting or to set this value *well* in excess of the number of jobs your system is likely to ever reach.

QACTJOB: Initial number of active jobs

This value specifies the initial number of active jobs for which storage is allocated during IPL. To determine a value for QACTJOB, review the Active column (7) shown in the In-use Entries section of the display, as shown in Figure 5-4 on page 96. Set the value high enough to avoid allocation of storage for QADLACTJ additional active jobs while production work is active. The present setting is shown on the DSPJOBTBL display (4), see Figure 5-3 on page 96.

The shipped value is 20. We suggest you set the value 20% higher than the number of active jobs observed on a heavy-use day.

Attention: Changes to the QACTJOB system value do not take affect until the next IPL.

QADLACTJ: Additional number of active jobs

This specifies the additional number of active jobs that have storage allocated when the initial number of active jobs (QACTJOB) is reached. Setting this value too low consumes system resources during production activity when many additional jobs are needed. Setting the number too high consumes resources allocating additional storage which may never be used. The present setting is shown on the DSPJOBTBL display (5), see Figure 5-3.

The shipped value is 10. We suggest you retain this default. A change to this system value takes effect immediately.

Storage system values

The allocation system values above typically do not change much when a Web serving workload is added to the system. This is because adding the HTTP and application servers adds only a modest number of jobs to the system workload. The following storage system values are typically more significant, because they need to account for the individual threads running within jobs. And adding a Web serving configuration to a system can easily add hundreds of threads to the system activity.

The number of threads that is allowed to run is associated with the storage system values because typically main storage contention would be the primary reason to limit the number of threads allowed to run concurrently. Such main storage contention between threads is often called *thrashing*. The term refers to a condition where main storage is overcommitted. The result is that the system spends considerable resources managing main storage page faults for the jobs that are running, and the jobs are stealing main storage from each other, thus furthering the need for the system to fault in more main storage pages.

To avoid thrashing, parameters are established to limit the number of threads that can actively use the CPU at the same time. Once the limit is reached, a thread that is ready to use the CPU is held in an ineligible state until another thread gives up the CPU to wait for disk I/O, user response, or other conditions to occur. By limiting the number of active threads, we avoid *potential* contention between them. However, if we set the limits too low, then the limit itself becomes the bottleneck, rather than main storage contention, or other system resources, thus leaving the system's resources under-utilized. The rest of this section shows you how to achieve a balance between overcommitted and under-utilized system resources.

To review all of the storage system values, use the following command:

```
WRKSYSVAL SYSVAL(*STG)
```

QBASACTLVL: Base storage pool activity level

This specifies the number of threads that are allowed to run concurrently in the *BASE main storage pool. By default, the QEJBAS5 subsystem used by WebSphere Application Server uses this *BASE pool. For information on moving the subsystem to another memory pool, see 5.3, "Configuring main storage pools" on page 99.

The shipped value is 6. If your QEJBAS5 subsystem is running in the *BASE pool, we suggest that this value be set to a minimum of 1500. A change to this system value takes effect immediately. This value is revised by the performance adjuster when system value QPFRADJ is set to 1, 2, or 3. However, the performance adjuster is often not able to react to sudden, dramatic changes in the number of active threads in the pool. Starting the application server or HTTP server powered by Apache is just such a sudden, dramatic change in the number of active threads.

QBASPOOL: Base storage pool minimum size

This specifies the *minimum* number of kilobytes of main storage allocated for the *BASE pool. The *BASE pool is given all main storage that is not specifically designated to other pools. But designating a value that is too high may not leave enough storage to create the other pools at their configured size.

The shipped value is 5% of the system's main storage with a minimum value of 2000KB. We suggest you retain this default value. A change to this system value takes effect immediately.

QMAXACTLVL: Maximum activity level of system

This specifies the number of *threads* that are allowed to be active system-wide in all of the active subsystems. Since this can be controlled by the individual pools as well, we suggest you limit the number of active threads at that level. For more information, see 5.3, "Configuring main storage pools" on page 99.

The shipped value is *NOMAX. We suggest you retain this default value. A change to this system value takes effect immediately

QMCHPOOL: Machine storage pool size

This specifies the initial size, in kilobytes, of the *MACHINE pool (pool 1 on the WRKSYSSTS display). Since this pool is where the most commonly used portions of the operating system and licensed programs are loaded, it is crucial that it be large enough. This pool should have the lowest fault rates on the machine. However, making it too large means that there is main storage in the machine that is not being utilized.

We suggest using the intermediate assistance level of the WRKSYSSTS command to review the number of DB Faults and Non-DB Faults shown. Use the F5 (Refresh) key to capture the average fault rates over a period of time. We suggest reviewing the figures once an elapsed time of at least 5 minutes is reached. Add the DB and Non-DB fault figures (note: *faults*, not the *pages*) together. If the result exceeds 10, then we suggest you increase the pool size by 10% from its present value. Once the value has been adjusted, use the F10 (Restart) key to begin a fresh collection of fault rates. Then reassess whether further increases are necessary.

The shipped value is 20000KB. We suggest increasing the pool size by 10% until a total faulting rate in the pool falls below 10. A change to this system value takes effect immediately. You may also alter the size of the machine pool directly from the WRKSYSSTS display, when using the intermediate or advanced assistance level of the display (use F21 to select the assistance level).

5.3 Configuring main storage pools

Main storage (memory) on the iSeries is segmented into pools. The subsystems that jobs run within are configured to perform their work in specific pools. By default, subsystems perform their work in the *BASE pool. However there can be advantages to separating main storage into additional pools, and designating which subsystems run in those pools. It gives you more granular control over the main storage devoted to specific types of work. This section discusses how to create main storage pools, how to designate which work is performed in them, and how to define whether or not the automated performance adjuster modifies the pool's size and activity level.

5.3.1 Creating a main storage pool

Creating a separate memory pool for a subsystem allows us to establish an area of main storage for a specific workload. This helps assure that the main storage is available when the jobs running in the pool require it. It also assures that the jobs only compete with each other for main storage in the pool. The disadvantage of creating separate pools is that other jobs cannot use the memory in the pool, even when the jobs it is designated for are not active.

The performance adjuster (QPFRADJ system value) attempts to balance things, by increasing the size of highly used pools, and gradually reducing the size of low use pools. This can mean that a pool designated for a specific task, such as application serving, may have the majority of its memory pulled away by the performance adjuster. This would occur if the application server is not servicing many clients, or if it is ended for an extended period of time. If a large number of requests begin coming in (when the business day starts, for example), the pool associated with the application server may thrash for quite some time until performance adjustment sufficiently increases the size of the pool.

The system supports up to 63 *shared* pools, which may be used by one or more subsystems. These shared pools exist all the time, although most have no memory associated with them. To create a shared pool, use the WRKSHRPOOL command (as shown in Figure 5-5), and enter a size greater than.25 and an appropriate activity level for the pool. The activity level represents the number of threads that are allowed to be in an active state at the same time. If the activity level is exceeded, threads that become ready to use the CPU after waiting for disk I/O or other activity to complete, are held in an inactive state (ineligible) until another thread goes into a wait state. The activity level and pool size of shared pools are adjusted periodically by the performance adjuster if the QPFRADJ system value is set to 2 or 3.

```
                          Work with Shared Pools
                                                              System:   AS07
Main storage size (M)  . :        7000.00

Type changes (if allowed), press Enter.

             Defined    Max   Allocated   Pool  -Paging Option--
Pool        Size (M)  Active  Size (M)     ID   Defined  Current
*MACHINE      950.00   +++++      950.00    1   *FIXED   *FIXED
*BASE        5579.75    2500     5579.75    2   *FIXED   *FIXED
*INTERACT     400.00     125      400.00    3   *CALC    *CALC
*SPOOL         70.00       5       70.00    4   *FIXED   *FIXED
*SHRPOOL1        .00       0                    *FIXED
*SHRPOOL2        .00       0                    *FIXED
*SHRPOOL3        .00       0                    *FIXED
*SHRPOOL4        .00       0                    *FIXED
*SHRPOOL5        .00       0                    *FIXED
*SHRPOOL6        .00       0                    *FIXED
                                                                More...
Command
===>
F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve   F11=Display tuning data
F12=Cancel
```

Figure 5-5 WRKSHRPOOL display

Once you've established a size and activity level for a shared pool (*SHRPOOL1 for example), then you must add this pool to the subsystem. Use the following command to associate the QEJBAS5 subsystem's second pool with *SHRPOOL1:

```
CHGSBSD SBSD(QEJBAS5/QEJBAS5) POOLS((2 *SHRPOOL1))
```

Private pools are used only by the subsystem that defines them, and exist only while the subsystem is active. Private pools are created when you perform a CHGSBSD command, and designate a POOL parameter with an integer value for the storage size, rather than one of the special values of *SHRPOOLnn. This value defines, in kilobytes, the amount of main storage to allocate exclusively for this subsystem. The minimum value is 256 (256KB). The activity level and pool size of private pools is *not* adjusted by the performance adjuster. Private pools are also not shown on the WRKSHRPOOL display. Use the WRKSYSSTS display to see them.

To create a private memory pool for the QEJBAS5 subsystem, you could use a command similar to the following:

```
CHGSBSD SBSD(QEJBAS5/QEJBAS5) POOLS((2 800 500))
```

This command creates a *private pool* of 800MB with an activity level of 500, and that pool becomes pool 2 for the QEJBAS5 subsystem. Pool 1 generally is set to *BASE, and it is where the subsystem monitor is run. We suggest you leave pool 1 of the subsystem set to *BASE.

5.3.2 Running subsystem QEJBAS5 in a separate memory pool

Once you've created a memory pool, and configured the QEJBAS5 subsystem to use it as its second pool, you must route work into this pool. This is done via the routing entries in the subsystem description. The QEJBAS5 subsystem only has one routing entry for all of its work. You can change that existing routing entry to direct all of its work into pool 2 of the subsystem, using the CHGRTGE command:

```
CHGRTGE SBSD(QEJBAS5/QEJBAS5) SEQNBR(9999) POOLID(2)
```

Note: POOLID is the number of the pool we designated for the subsystem on the CHGSBSD command. This is a different concept from the shared pool identifier. For example, if you use the shared pool example 5.3.1, "Creating a main storage pool" on page 100, we've configured the subsystem's POOLID(2) to correspond to *SHRPOOL1. This also differs from the pool ID that shown on the WRKSYSSTS display. See 5.3.3, "Adjusting main storage pools manually" on page 101 for more information about how the pool IDs interrelate.

The new routing entry takes affect the next time the QEJBAS5 subsystem is started.

5.3.3 Adjusting main storage pools manually

Once you've established your pool and subsystem configuration, you should periodically verify that it is running efficiently. Over time, numbers of users, and the mix of applications which are run, may vary. This section describes how to monitor your system, and how to know when adjustments may help improve performance.

Reviewing main storage pool size

When a memory pool is too small, the tasks running in that pool are pulling in information they require, and thus forcing out memory pages of other tasks. The result is that before any given task can get much done, it must bring back memory pages it requires to complete its work. One symptom of an undersized memory pool is poor user response times, with *low* CPU utilization. To confirm if page faulting is the cause of poor response times, do the following:

- Determine system pool of jobs with slow response times.
- Determine if the size of the system pool should be increased.

Determining system pool of jobs with slow response times

Use the WRKACTJOB command, then press F11 (Display elapsed data) once, as shown in Figure 5-6. The system pool ID (1) is the fourth column of the display. As you see in the example, the jobs are all running in system pool 5. If the jobs are not running in the pool you desire, confirm that the routing entry for the QEJBAS5 subsystem is directing the work to the desired pool, as discussed in 5.3.2, “Running subsystem QEJBAS5 in a separate memory pool” on page 101.

```
                          Work with Active Jobs                    AS07
                                                        12/02/02  10:04:31
CPU %:    1.3     Elapsed time:   00:04:42     Active jobs:   245

Type options, press Enter.
  2=Change   3=Hold   4=End   5=Work with   6=Release   7=Display message
  8=Work with spooled files   13=Disconnect ...
                                            --------Elapsed---------
Opt  Subsystem/Job  Type  Pool1 Pty      CPU  Int    Rsp  AuxIO  CPU %
     QEJBAS5        SBS      5   0        .0                    0     .0
       JMSSAMPLE    BCH      5  25+     225.6                    0     .0
       MYSERVER     BCH      5  25+     439.6                    0     .0
       QEJBMQLSR    BCI      5  25        .0                    0     .0
       SAMPLE       BCH      5  25+    1059.1                    0     .2
       SERVER1      ASJ      5  25+     572.8                    0     .0
       WEBFACE      BCH      5  25+    1265.4                    0     .1


                                                                  Bottom
Parameters or command
===>
F3=Exit   F5=Refresh      F7=Find      F10=Restart statistics
F11=Display thread data   F12=Cancel   F23=More options   F24=More keys
```

Figure 5-6 WRKACTJOB SBS(QEJBAS5) to confirm job’s pool ID

Determining if the size of a system pool should be increased

To see if system pool 5 has high faulting rates, use the WRKSYSSTS command as shown in Figure 5-7. Note that the Elapsed time (1) in our sample display is only 1 second, we should wait for a few minutes, and use the F5 (Refresh) key to refresh the display before deciding about necessary adjustments. A time interval of at least five minutes gives us a better picture of overall system performance, rather than just a brief point in time. If your WRKSYSSTS display does not resemble Figure 5-7, use the F21 key to change your assistance level to 2 (Intermediate) or 3 (Advanced). Note that at the time of our example display, there was no performance problem in system pool 5.

```
                          Work with System Status                   AS07
                                                          12/02/02  09:36:24
% CPU used . . . . . . . :         .5    Auxiliary storage:
% DB capability  . . . . :         .0      System ASP . . . . . . :    52.74 G
Elapsed time . . . . . . :   00:00:01 1    % system ASP used  . . :    61.3955
Jobs in system . . . . . :         951      Total  . . . . . . . . :    52.74 G
% perm addresses . . . . :        .012      Current unprotect used :     7760 M
% temp addresses . . . . :        .012      Maximum unprotect  . . :     8124 M

Type changes (if allowed), press Enter.

System    Pool    Reserved    Max   ------DB------   ----Non-DB---
 Pool   Size (M)  Size (M)  Active  Fault   Pages   Fault   Pages
   1      950.00    336.50   +++++     .0     .0       .0      .0
   2     4780.00      2.83    2500     .0     .0       .0      .0
   3      400.00       .01     125     .0     .0       .9      .9
   4       70.00       .00       5     .0     .0       .0      .0
   5      800.00       .00     500     .0 2   .0       .0 3    .0
                                                                Bottom
Command
===>
F3=Exit   F4=Prompt          F5=Refresh   F9=Retrieve   F10=Restart
F11=Display transition data  F12=Cancel   F24=More keys
```

Figure 5-7 WRKSYSSTS to review page faulting rates

If the fault rates (add 2 and 3) in system pool 5 were markedly higher than in the other system pools, or if the other system activity could accept a performance degradation, then we could increase the size of pool 5 by entering a new pool size on the WRKSYSSTS display. The system then reduces the size of the *BASE pool by the amount of your change. This may result in an increase in the faulting rates and response times of the jobs running in the *BASE pool.

In general, you should make 10% adjustments to pool sizes until the faulting rates are balanced. In our example, if pool 5 had higher faulting rates than the other pools, we would increase it to a size of 880MB. Then use F10 to restart gathering statistics, wait about 5 minutes, and press F5 to refresh the display. Then reassess if further adjustment is required.

If the faulting rates in *all* the memory pools are high, then purchasing additional main storage, or rescheduling some of the activity, would likely improve the response times observed when running this workload. In such cases, additional main storage also reduces the total disk activity on the system, which may reduce contention for disk arms on the system, and further improve performance.

Reviewing memory pool activity level

Once you have confirmed that the memory pool *sizes* are reasonable for your workload, you may find that the faulting rates are still higher than you would like. If you are not able to devote additional main storage to the pool, then *activity levels* may help you streamline the work done in the pool.

The activity level appropriate for your pools varies depending upon your application. Use the following steps to assess and adjust the activity levels you have established for your memory pools.

The basic idea of activity levels is that putting an upper limit on the number of concurrent threads, increases the likelihood that a thread is able to complete its work and come to a normal wait state, before being interrupted. With that thread no longer requiring CPU time, the system can devote its attention to another thread without having to handle both of them at the same time. By reducing the number of threads that are actually allowed to use the CPU concurrently, we hope to reduce the contention for memory in the pool. And thus reduce page faulting and improve efficiency.

Threads are considered to be in one of the following states at all times:

- Active
- Wait
- Ineligible

An *active* thread is one that has all the resources it presently requires to use the CPU. The actual CPU time devoted to such a thread depends upon priority, and timeslice. Once it is given some CPU time, the thread may access its heap storage or some other object that is not presently in main storage. This causes a page fault to get the required memory pages. While the thread is waiting for this to occur, it transitions to *wait* state.

A thread remains in a *wait* state until the resource it was waiting for becomes available. Note that at the level of this discussion, a wait state may not be reflected in the status of the job of thread on the WRKACTJOB, and DSPJOB displays. Those displays are showing waits on mutex or sockets data which are longer in duration. The thread also waits for page faults to be processed, internal seizes, and other short duration events. Once the resource or event that was needed is available, the thread is again ready to use the CPU. Before it may do so, it must allocate an available activity level in the pool. If an activity level is available, the thread transitions back to an *active* state. If no activity levels are presently available, then the thread transitions to *ineligible* state.

A thread remains in an *ineligible* state, and is unable to use the CPU, until another thread in the pool reaches a *wait* state for some reason, thus releasing an available activity level. When an activity level becomes available, the thread transitions from *ineligible* to *active* state. This is how the activity level can streamline the work performed in the pool.

To review the thread transitions between states in the various main storage pools on the system, use the WRKSYSSTS command, and press F11 (Display transition data). The resulting display is shown Figure 5-8.

```
                          Work with System Status                   AS07
                                                          12/02/02  12:39:07
% CPU used . . . . . :        1.7    Auxiliary storage:
% DB capability  . . . :        .0      System ASP . . . . . :    52.74 G
Elapsed time . . . . . :   00:00:05      % system ASP used  . . :    60.9188
Jobs in system . . . . :        961      Total  . . . . . :    52.74 G
% perm addresses . . . . :      .012      Current unprotect used :     7499 M
% temp addresses . . . . :      .012      Maximum unprotect  . . :     8124 M

Type changes (if allowed), press Enter.

System    Pool    Reserved    Max   Active->  Wait->  Active->
 Pool   Size (M)  Size (M)  Active    Wait     Inel     Inel
   1      950.00    336.56   +++++    23.6       .0       .0
   2     4780.00      2.82    2500    3618       .0       .0
   3      400.00       .01     125    11.8       .0       .0
   4       70.00       .00       5      .0       .0       .0
   5      800.00       .00     500 1    .0       .0       .0
                                                          Bottom
Command
===>
F3=Exit   F4=Prompt     F5=Refresh   F9=Retrieve   F10=Restart
F11=Display pool data   F12=Cancel   F14=Work with subsystems   F24=More keys
```

Figure 5-8 WRKSYSSTS transition data

The last three columns present the number of transitions per minute, on average, that have occurred over the elapsed time shown in the upper left. As you can see, transitions from *active* to *wait* are the most common (see the Active->Wait column of the display). The last two columns present the threads that have gone to an *ineligible* state during the elapsed time.

Determining if activity level of memory pool should be changed

We examined system pool 5 since we configured our WebSphere applications to run there. As you can see from the display, this pool was idle on our system. If you observed high page faulting when determining the pool size, and were unable to increase the pool size, then you may want to *reduce* the activity level of the pool. This is shown in the Max Active column of the display (1), and may be changed by typing a new value and pressing Enter, then pressing F10 to reset the values being displayed.

Another possible cause of low throughputs or slow response times is that the activity level may be too *low*. When this occurs, resources are available for a thread to get work done, but it is not allowed to run, because it is waiting for an activity level. The activity level itself becomes the bottleneck, rather than the pool's available memory or the CPU. After you have confirmed a low page faulting rate (as shown in "Reviewing main storage pool size" on page 101), you may find that increasing the activity level more fully utilizes the system resources, and provides better throughputs and response times.

6

WebSphere Application Server 5.0: configuration and administration

In this chapter we describe the steps you need to perform after the installation of WebSphere Application Server V5.0 for iSeries.

The first part of the chapter covers the setup of the WAS environment on iSeries with a default instance and a default application server, and verification of this installation, starting with 6.2, “Setting up the WebSphere Application base environment” on page 109.

If you don’t want to use the default instance when you start the WAS environment (for example, if you only want to work with your individual instances and application servers), then you have to change the WAS subsystem description by removing the autostart job. This is described in “Changing the default behavior for the QEJBAS5 subsystem” on page 113.

The second part of the chapter covers additional functions like creating and working with multiple WebSphere Application server instances. The creation of an individual instance is described in 6.5.2, “Multiple instances of WebSphere Application Server” on page 113.

We show by examples how to create and administrating additional resources:

- With the bank application example, we show how to set up the JDBC providers that your applications can use to access a database; see 6.13.1, “Administering JDBC providers and DataSources” on page 187.
- With the MDBsample application, we show how to set up the embedded Java Message Service (JMS) to exchange data asynchronous between a JMS client and a message driven bean (MDB). We also describe how to set up the JMS resources and how to deploy and test the MDBsample application; see 6.13.3, “JMS Administration in WebSphere 5.0” on page 220.

6.1 QShell scripts used in our scenario

IBM WebSphere Application Server provides several scripts that can be used via the QShell interface of the iSeries server for administration of the WAS instances and servers.

Table 6-1 shows some of the scripts which are available for the WAS environment. In the column *pointer* we refer to the sections that describe how to invoke these scripts.

Instead of using the QShell scripts, you can also use the wsadmin tool (see Chapter 9, “The wsadmin tool” on page 385), or you can use the administrative console to do the desired configuration tasks.

Table 6-1 QShell scripts

Command	Usage	Pointer	Syntax reference
Script commands used for an instance/server in the WAS and ND environment			
crtwasinst	Create WAS instance	“Creating a new WAS instance” on page 113	A.7.1, “Syntax and parameters for crtwasinst script” on page 523
dspwasinst	Display WAS instance	6.5.8, “Displaying a WAS instance via the dspwasinst script” on page 129 6.11.1, “Creating an additional application server to an instance” on page 180	
chgwassvr	Change WAS instance/server	6.5.7, “Changing a WAS application server via the script” on page 128	
dltwasinst	Delete WAS instance	6.5.9, “Deleting a WebSphere Application Server instance” on page 131	A.7.5, “Syntax and parameters for dltwasinst script” on page 538
startServer	Start WAS instance/server	6.5.3, “Starting a specific application server” on page 117	A.7.2, “Syntax and parameters for startServer script” on page 532
stopServer	Stop WAS instance/server	6.5.6, “Stopping an application server” on page 124	A.7.3, “Syntax and parameters for stopServer script” on page 533
Script commands used from the WAS environment to administer nodes in an ND environment			
addNode	Add node to cell	7.3.2, “Adding a node to the network deployment instance” on page 280	A.7.8, “The syntax of the addNode script” on page 541
startNode	Start nodeagent	“Starting a node agent via startNode script” on page 291	A.7.9, “The syntax of the startNode script” on page 543
stopNode	Stop nodeagent	“Stopping a node agent via stopNode script” on page 295	

Command	Usage	Pointer	Syntax reference
syncNode	Synchronize node with cell	“Using the syncNode script” on page 290 “Synchronizing the configuration via the ND administrative console” on page 306	
removeNode	Remove node	“The removeNode script” on page 296	
cleanupNode		“Cleanup node” on page 297	
Script commands used from the ND environment only			
startManager	Start the deployment manager for an ND instance	“The startManager script” on page 279	A.7.6, “Syntax and parameters of the startManager script” on page 538
stopManager	Stop the deployment manager for an ND instance	“The stopManager script” on page 279	A.7.7, “Syntax and parameters for the stopManager script” on page 539

6.2 Setting up the WebSphere Application base environment

Once the WebSphere Application Server V5.0 for iSeries software and the group PTFs for WebSphere Application Server have been installed on the iSeries server, it is now time to start the WebSphere Application Server environment and to verify the installation by testing it.

The WebSphere Application Server 5.0 product includes a default instance with the name default that consists of a single application server. The application server, named server1, contains the administrative console application, a default application, and the install verification application. These components are created at install time. They are started each time you start the WAS subsystem. Also, the needed configuration files that are arranged in a set of cascading directories for the WAS environment are created at product install time as well, and they are stored in the Integrated File System (IFS) on iSeries. See 2.18, “IFS directory structure” on page 46.

This default instance and the application server, server1, can be used for your production environment. You can deploy your applications to this server and configure additional resources you need to support these applications (like the JDBC providers and JMS resources). To work with the WAS embedded JMS, you have to enable JMS for the default instance using the `chgwassvr` script, because JMS is not enabled to this instance by default. For details on how to do this; see “Changing a WAS application server via the script” on page 128 and Figure 6-18 on page 129.

WebSphere Application Server provides an internal HTTP server. You can use this internal HTTP server to verify the WebSphere Application Server configuration. This is described in “Verifying installation using one of the sample Web applications” on page 122.

For a production environment, an external HTTP server is recommended. In 6.6, “Configuring an HTTP server instance” on page 132, we explain how to create and configure an HTTP server instance.

Before you can use and start the WebSphere Application Server, ensure that the iSeries environment is set up correct. Ensure that following steps are already done:

1. Configure TCP/IP on iSeries server — see 2.9, “Starting, configuring, and verifying TCP/IP” on page 21, and be sure that TCP/IP is already started.
2. Set SQL server jobs — see 5.1.1, “Configuring MAXJOBS for database servers” on page 88.

6.3 Quick start tutorial

Follow the steps in Table 6-2 in sequence, to start the WebSphere Application Server base environment and to verify the installation.

Table 6-2 Steps for starting the WebSphere Application Server environment

Step		Chapter	Page
1		Configuring software license information	111
2		Special configurations: If you don’t want to use the default instance with the default application server, the special steps described below must be done before step 3: 1. Delete the autostart job entry from the WAS subsystem; see “Changing the default behavior for the QEJBAS5 subsystem” on page 113. 2. Create a new instance with different ports according to the default ports for all the services; see 6.5.2, “Multiple instances of WebSphere Application Server” on page 113. 3. Start the WAS subsystem, use the STRSBS QEJBAS5/QEJBAS5 command from an iSeries command line, and press Enter. This is similar to step 3 in this table, but without creating and starting a default instance. 4. Start the new instance; see “Starting a specific application server” on page 117. 5. Verify that your instance is started; see 6.5.4, “Verifying that the WAS environment has started” on page 119. 6. Continue beginning with step 5 of this table.	
3		Starting the WebSphere Application Server environment	112
4		Verifying that the WAS environment has started	119
5		Invoking the install verification script	122
6		Verifying installation using one of the sample Web applications	122
7		By default, the default instance is not enabled for using JMS. If you want to work with the WAS embedded JMS in your default instance, enable JMS by using the `chgwasinst` script; see Figure 6-18 on page 129.	129
8		Configure HTTP Server; see Configuring an HTTP server instance: If you want to use the IBM HTTP Server (powered by Apache) do step 8a, and depending on the OS/400 Version of your iSeries server, do step 8b or 8c. If you want to use the Domino Web server, do step 8d.	132
	a	Configuring IBM HTTP Server (powered by Apache) in OS/400 V5R1	133
	b	Configuring IBM HTTP Server (powered by Apache) in OS/400 V5R2	143
	c	Configuring Lotus Domino Web server	157
9		Working with the Administrative Console	159
10		Configuring a virtual host	161

Step		Chapter	Page
11		Updating Web server plug-in configuration	171
12		Restarting the WebSphere Application Server	176
13		Starting IBM HTTP Server for iSeries	176
14		Verifying installation using the external IBM HTTP server	179

6.4 Configuring software license information

Before you start the WAS environment for the first time, you need to configure a software license information for WAS:

1. Enter the Work with License Information (WRKLICINF) command on an OS/400 command line and press Enter.
2. On the Work with License Information menu, press F11 (Display Usage Information).
3. Scroll the page and move the cursor to the line that contains the product 5733WS5, Feature 5050.
4. Select option **2** (Change), and press Enter.
5. On the Work with License Information (CHGLICINF) display, press F9 (all parameters). This brings up the screen that is shown in Figure 6-1.
6. Update the Usage limit (USGLMT) value with the usage limit shown on Proof of Entitlement (POE) or invoice. Entering any number that exceeds the purchased limit violates the IBM purchase agreement.
7. Update the Threshold (THRESHOLD) value with ***USGLMT**, and press Enter. Do not leave the threshold set to zero.

Note: You may see the following message:

```
CPA9E1B: Usage limit increase must be authorized.
```

Respond by typing `G` and pressing Enter.

```
                    Change License Information (CHGLICINF)

Type choices, press Enter.

Product identifier . . . . . > 5733WS5       Identifier
License term . . . . . > V5            Vx, VxRy, VxRyMz, *ONLY
Feature  . . . . . > 5050          5001-9999
Usage limit  . . . . . .   *NOMAX        0-999999, *SAME, *NOMAX...
Alternate usage limit:
  Identified usage . . . . . .   0             0-999999, *SAME
  Unidentified usage . . . . . .   0           0-999999, *SAME
Threshold  . . . . . .   *USGLMT       0-999999, *SAME, *CALC...
Message queue  . . . . . .   *NONE         Name, *SAME, *NONE, *OPSYS
  Library  . . . . . .                     Name, *LIBL, *CURLIB
               + for more values

Log  . . . . . .   *NO           *SAME, *NO, *YES




                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 6-1 Change License Information

6.5 Basic administrative tasks

After you have installed WebSphere Application Server V5.0 for iSeries, you need to master basic administration skills. This section describes them in the following sequence:

- Starting the WebSphere Application Server environment
- Creating a new WAS instance
- Starting a specific application server
- Verifying the installation
- Stopping an application server
- Displaying and changing an application server
- Deleting a WAS instance

6.5.1 Starting the WebSphere Application Server environment

WebSphere Application Server jobs run in subsystem QEJBAS5. By default, the subsystem QEJBAS5 has an autostart job entry (AJE) that starts the default instance with the default application server (SERVER1). So when you start the QEJBAS5 subsystem, the default instance is started automatically.

A user profile with *JOBCTL authority is required to start the WebSphere Application Server environment.

To start the QEJBAS5 subsystem, enter this command on an OS/400 command line and press Enter.

```
STRSBS QEJBAS5/QEJBAS5
```

Starting the WAS subsystem automatically at iSeries system startup

You can configure iSeries in such a way that the QEJBAS5 subsystem starts automatically at iSeries system startup.

Note: TCP/IP must be active before WebSphere Application Server environment can start. Ensure that the STRTCP command runs before the STRSBS QEJBAS5/QEJBAS5 command in your startup program or in your autostart job.

In order to achieve this, add an autostart job entry to the QSYSWRK subsystem:

1. Enter this command on an iSeries command line and press Enter:

   ```
   ADDAJE SBSD(QSYSWRK) JOB(QEJBSTART) JOBD(QEJBAS5/QEJBSTART)
   ```

2. You will see the following message:

   ```
   Change effective next time subsystem starts.
   ```

3. If an autostart job entry already exists, you will get this message.

   ```
   Autostart job entry already exists for job QEJBSTART.
   ```

Changing the default behavior for the QEJBAS5 subsystem

If you do not want to start the default application server (server1) at startup time, change the QEJBAS5 subsystem description before starting it for the first time.

You can achieve this by deleting the autostart job entry from the QEJBAS5 subsystem. Follow these steps:

1. Remove the autostart job entry by running the remove autostart job entry (**rmvaje**) command on an iSeries command line and pressing Enter:

   ```
   rmvaje sbsd(qejbas5/qejbas5) job(server1)
   ```

   Here, `server1` is the name of the default application server.

2. You will get this message:

   ```
   Change effective next time subsystem starts.
   ```

6.5.2 Multiple instances of WebSphere Application Server

Multiple instances of WebSphere Application Server can run concurrently on an iSeries server. The use of multiple instances allows you to create multiple WebSphere Application Server environments (instances) that are completely isolated from one another on the same system. Every WAS instance runs in its own Java Virtual Machine (JVM). Working with multiple instances can be very useful when you want to run multiple environments. For example, you can create separate instances for application development and application testing, or you can create one instance with security enabled and one with security disabled.

Creating a new WAS instance

To create a new instance, you need to run the `crtwasinst` script in QShell. This script creates all new server directories and configuration files in IFS and sets up the correct authorities. For more information about the directory structure and the most important files, refer to 2.18, “IFS directory structure” on page 46. This script is available in both WebSphere Application Server and WebSphere Application Server Network Deployment.

The `crtwasinst` script is located in the WAS_INSTALL_ROOT/bin directory where WAS_INSTALL_ROOT is located:

- `/QIBM/ProdData/WebAS5/Base/bin` for WebSphere Application Server V5.0 for iSeries

- `/QIBM/ProdData/WebAS5/ND/bin` for WebSphere Application Server Network Deployment V5.0 for iSeries

If you use the script from the Network Deployment directory, you create a new Network Deployment instance. This is a Deployment Manager instance, not an application server instance. For more information about Network Deployment, refer to Chapter 7, "WebSphere Application Server Network Deployment 5.0: configuration and administration" on page 263.

To run this script, your iSeries user profile must have *ALLOBJ authority.

Note: It is important that you know which ports are already in use before running this script. Each WAS script requires 12 TCP/IP ports.

To create a new WAS instance, follow these steps:

1. On the OS/400 command line, enter the STRQSH (Start QShell) command and press Enter.
2. On the QShell command line, use the **cd** command to change to the directory that contains the script:

   ```
   cd /QIBM/ProdData/WebAS5/Base/bin
   ```

3. Type `crtwasinst` with the appropriate parameters (for more information about the parameters; see A.7.1, "Syntax and parameters for crtwasinst script" on page 523). In our examples we use the most typical parameters:

   a. Example 1:

      ```
      crtwasinst -instance webface -portblock 10400 -noembeddedjms
      ```

      Here is a description of the parameters:

      - `-instance`: This is a required parameter. It is used to specify the new WAS instance's name. In our example - webface.
      - `-portblock`: This is an optional parameter. However, we recommend that you use it all the time. It allows you to allocate 12 consecutive ports for the new WAS instance. `portblock` parameter specifies the first port number out of 12. In our example, crtwasinst will allocate the block of port from 10400 to 10411 for the new WAS instance.
      - `-noembeddedjms`: This is an optional parameter. Because we don't plan to use the embedded JMS server in this application server, we use the `-noembeddedjms` parameter. When this parameter is specified, the instance is *not* enabled to use the embedded JMS server.

      The result of running this script will be a new WAS instance, `webface`, with a single application server named `webface` (see Figure 6-2).

      Notice that the external HTTP port is set to the default HTTP port 80.

```
crtwasinst -instance webface -portblock 10400 -no embeddedjms
Creating instance webface...
Instance webface created.
Instance root directory is /QIBM/UserData/WebAS5/Base/webface.
The application server name is webface.
Ports:
 External HTTP:  80
  External HTTPS: 443
  Name service:   10400
  JMS secure:     10401
  JMS queued:     10402
  JMS direct:     10403
  DRS client:     10404
  SOAP:           10405
  SAS:            10406
  CSIV2 Mutual:   10407
  CSIV2 Server:   10408
  Internal HTTP:  10409
  Admin:          10410
  Admin SSL:      10411
$
```

Figure 6-2 crtwasinst message for instance webface

b. Example 2:

```
crtwasinst -instance jmsmdb -exthttp 10400 -inthttp 10401 -admin 10402 -portblock
10405
```

We used additional parameters as compared to the previous example:

- `-exthttp`: This is an optional parameter. It specifies the external HTTP port for the new WAS instance.
- `-inthttp`: This is an optional parameter. It specifies the internal HTTP port for the new WAS instance. This parameter overrides the value that could be assigned when using the `portblock` parameter.
- `-admin`: This is an optional parameter. It specifies the HTTP port on which the administrative console is listening. This parameter overrides the value that could be assigned when using the `portblock` parameter.

The messages we get in the QShell panel is shown in Figure 6-3 on page 116. It also shows which TCP/IP port values are assigned for the different WAS services.

```
crtwasinst -instance jmsmdb -exthttp 10400 -inthttp 10401 -admin 10402 -portblock 10405
Creating instance jmsmdb...
ADCP0005I: Using cell RCHAS07_jmsmdb, node RCHAS07_jmsmdb and server jmsmdb.
ADCP0006I: Embedded JMS enabled.
Instance jmsmdb created.
 Instance root directory is /QIBM/UserData/WebAS5/Base/jmsmdb.
 The application server name is jmsmdb.
 Ports:
   External HTTP:  10400
   External HTTPS: 443
   Name service:   10405
    JMS secure:     10406
    JMS queued:     10407
    JMS direct:     10408
    DRS client:     10409
    SOAP:           10410
    SAS:            10411
    CSIV2 Mutual:   10412
    CSIV2 Server:   10413
    Internal HTTP:  10401
    Admin:          10402
    Admin SSL:      10414
    CSIV2 Server:   10413
    Internal HTTP:  10401
    Admin:          10402
    Admin SSL:      10414
 $
```

Figure 6-3 crtwasinst message for instance jmsmdb

c. Example 3:

```
crtwasinst -instance team -server uschi -portblock 10505 -exthttp 10500 -inthttp
10501 -admin 10502 -nodefaultapps
```

We used additional parameters as compared to the previous examples:

- `-server`: This is an optional parameter. It specifies the name of the new application server. If you omit this parameter, `crtwasinst` creates a server with the same name as the instance name.
- `-nodefaultapps`: This is an optional parameter. It allows you to skip the installation of the default sample applications to the new application server.

Figure 6-4 shows the messages in QShell after the script completes.

```
 crtwasinst -instance team -server uschi -portblock 10305  -exthttp 10300 -inthttp 10301
-admin 10302 -nodefaultapps
 Creating instance team...
 ADCP0005I: Using cell RCHAS02B_team, node RCHAS02B_team and server uschi.
 ADCP0006I: Embedded JMS enabled.
 Instance team created.
Instance root directory is /QIBM/UserData/WebAS5/Base/team.
The application server name is uschi.
 Ports:
   External HTTP:  10300
   External HTTPS: 443
   Name service:   10305
   JMS secure:     10306
   JMS queued:     10307
   JMS direct:     10308
   DRS client:     10309
   SOAP:           10310
   SAS:            10311
   CSIV2 Mutual:   10312
   CSIV2 Server:   10313
   Internal HTTP:  10301
   Admin:          10302
   Admin SSL:      10314
 $
```

Figure 6-4 Crtwasinst server uschi in instance team

4. After you create an instance, start the instance.

6.5.3 Starting a specific application server

Here we describe how you can start an application server for a specific instance, in two ways:

1. Run the `startServer` script from a QShell command line.
2. Start an application server when the subsystem QEJBAS5 is started; see “Starting a WAS instance and application server automatically” on page 119.

The `startServer` script reads the configuration file for the specified server process and starts the server. This script is available in both WebSphere Application Base Server and WebSphere Application Server Network Deployment. The `startServer` script is located in the WAS_INSTALL_ROOT/bin directory where WAS_INSTALL_ROOT is located:

- `/QIBM/ProdData/WebAS5/Base/bin` for WebSphere Application Server V5.0 for iSeries.
- `/QIBM/ProdData/WebAS5/ND/bin` for WebSphere Application Server Network Deployment V5.0 for iSeries.

If you use the script from the Network Deployment directory, you start a Network Deployment instance. This is a Deployment Manager instance, not an application server instance. For more information about Network Deployment, refer to Chapter 7, “WebSphere Application Server Network Deployment 5.0: configuration and administration” on page 263.

Running the startServer script

To run the `startServer` script, your iSeries user profile must have *ALLOBJ authority.

For information about syntax and available parameters for the `startServer` script; see A.7.2, “Syntax and parameters for startServer script” on page 532.

To start a WAS instance, follow these steps:

1. On the OS/400 command line, enter the STRQSH (Start QShell) command and press Enter.
2. On the QShell command line, use the **cd** command to change to the directory that contains the script, in our case for WebSphere Application Base Server:

   ```
   cd /QIBM/ProdData/WebAS5/Base/bin
   ```

3. Execute the following command:

   ```
   startServer -instance instance server
   ```

   Here, *instance* is the name of the WAS instance, and *server* is the name of the application server you're trying to start.

   For example:

   a. To start the server named `myServer` in the instance `myInstance`, you will execute the following command:

      ```
      startServer -instance myInstance myServer
      ```

Important: Server name is case-sensitive. Make sure you use the correct case.

   b. To start the default application server `server1` in the default instance we type:

      ```
      startServer server1
      ```

      If you omit the `instance` parameter, startServer will attempt to start the specified server in the `default` instance.

      You should see a message similar to the one shown in Figure 6-5.

```
startServer server1
CPC1221: Job 033139/QEJBSVR/SERVER1 submitted to job queue QEJBJOBQ in
library QEJBAS5.
EJB6123: Application server started.
 Cause . . . . . :   Application server server1 in Base instance default
   has started and is ready to accept connections on admin port 9090.
$
```

Figure 6-5 startServer for default server

   c. If you omit the server parameter, the script will attempt to start an application server which has the same name as the instance name. To start our instance `webface`, run the following command:

      ```
      startServer -instance webface
      ```

      In this example, startServer will attempt to start an application server with name `webface` in the instance `webface`.

Note: For the default instance, the `startServer` script uses *server1* for the server name parameter.

   d. To start the `jmsmdb` WAS instance:

      ```
      startServer -instance jmsmdb jmsmdb
      ```

   e. If you omit the instance parameter, the `startServer` script will attempt to start the specified application server in the `default` instance. If the name the server is not in the `default` instance, you'll see the error message:

```
startServer jmsample
```

You will get a message as shown in Figure 6-6.

```
> startServer jmsample
  CPC1221: Job 049599/QEJBSVR/JMSAMPLE submitted to job queue QEJBJOBQ in
  library QEJBAS5.
  EJB6121: Application server did not start.
   Cause . . . . . :   Application server jmsample in Base instance default
     failed to start. Check the joblog for job JMSAMPLE  QEJBSVR   049599 or
     the log files located in the instance's logs directory for more
     information.
  $
```

Figure 6-6 Application server not started

Starting a WAS instance and application server automatically

To start an individual server of an additional WAS instance when the subsystem QEJBAS5 is started, create a job description for that application server and add an autostart job entry to the QEJBAS5 subsystem.

The following steps describe how we create a job description for the instance named `webface`:

1. Create the job description with the CRTJOBD command from an iSeries command line. See Figure 6-7 as an example. `RQSTDTA` defines the iSeries program that will be called; it is QEJBSTRSVR for WAS.

   For the -instance parameter, specify the root directory of your instance, in our case it is `/QIBM/UserData/WebAS5/Base/webface`.

Note: This command has been wrapped for display purposes. Enter it on one line with spaces between the wrapped lines.

```
CRTJOBD JOBD(QEJBAS5/STRWEBFACE) JOBQ(QEJBAS5/QEJBJOBQ) TEXT('jobd for start WAS server
webface') USER(QEJBSVR) RQSDTA('QSYS/CALL PGM(QEJBAS5/QEJBSTRSVR) PARM(''-instance''
''/QIBM/UserData/WebAS5/Base/webface'' ''-server'' ''webface'')') LOG(4 0 *SECLVL)
```

Figure 6-7 JOBD for server webface

2. You should see a message similar to this one:

   ```
   Job description STRWEBFACE created in library QEJBAS5.
   ```

3. Add an autostart job entry for your new job by using the iSeries command ADDAJE:

   ```
   ADDAJE SBSD(QEJBAS5/QEJBAS5) JOB(WEBFACE) JOBD(QEJBAS5/STRWEBFACE)
   ```

6.5.4 Verifying that the WAS environment has started

Before you start the administrative console or before your WebSphere Application Server can process a client request, you should verify that the environment has been started successfully.

When the WebSphere Application Server environment is ready for use, a message is written to the job log of the application server job, indicating that the WebSphere Application Server environment is ready. It may take up to 20 minutes for the message to be displayed. It

depends on your iSeries server hardware configuration and workload. If the message is not displayed, refer to see Chapter 11, “Troubleshooting” on page 431.

Follow these steps to verify that the WebSphere Application Server environment for your instance has been started:

1. Run the Work with Active Jobs (WRKACTJOB) command on an OS/400 command line, specifying the QEJBAS5 subsystem on the subsystem (SBS) parameter and press Enter:

   ```
   WRKACTJOB SBS(QEJBAS5).
   ```

2. Find your application server job (the application server job for the default WebSphere Application Server instance is named SERVER1, as shown in Figure 6-8).

   If you have enabled your default WAS instance to use embedded JMS, the MQ listener job and several other MQ jobs in the QMQM subsystem are also started when your application server starts. You will see your application server job and an additional QEJBMQLSR job (the MQ listener job) in the QEJBAS5 subsystem.

   Note: The TEDHEAD application server runs with a different user profile TESTID19 (not the default user profile QEJBSVR). For more information about running the application server under a different user profile; see step e on page 174.

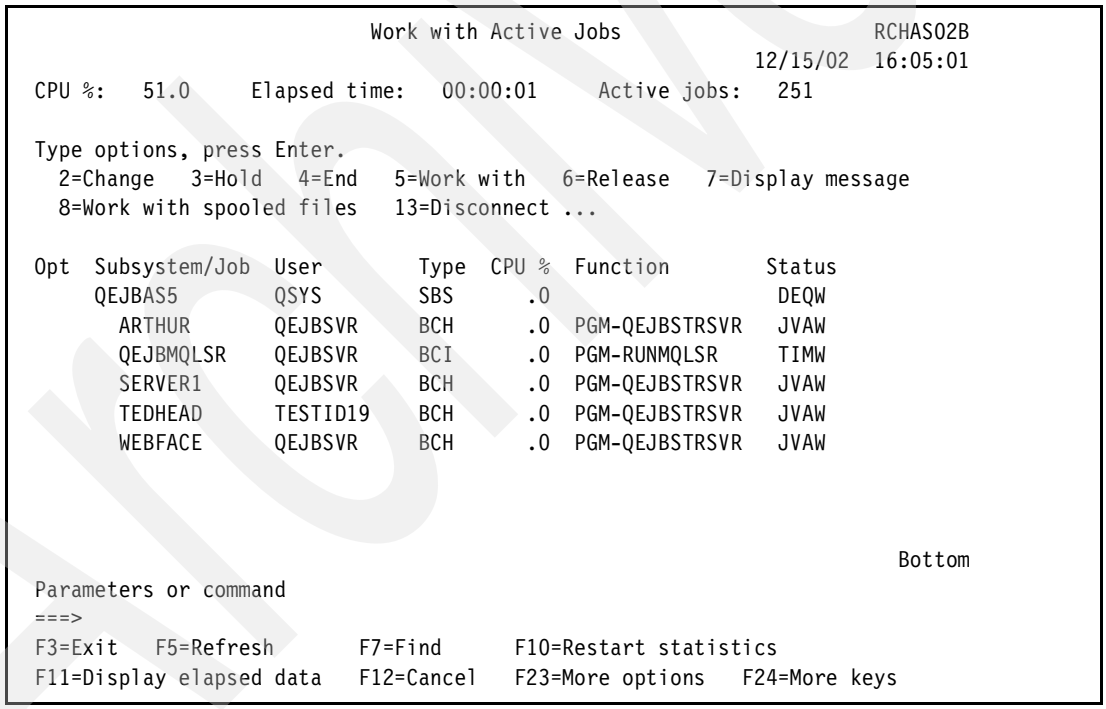


Figure 6-8 Work with active jobs

3. Specify option **5** (Work with Job) on the option line next to the application server job, and press Enter.
4. On the command line of the Work with Job display, type **10** (Display job log, if active), and press Enter.
5. Press F10 to display all messages.
6. Look for the message:

   ```
   WebSphere application server application_server ready
   ```

Here, `application_server` is the name of your application server. This message indicates that the application server is ready for use.

If the message is not displayed, press F5 to refresh the job log messages until the message is displayed.

7. To display the port number on which the administrative console is listening, position the cursor under the message 6 and press F1.

   The Additional Message Information display shows the port number (see Figure 6-9).

```
                        Additional Message Information

Message ID . . . . . :   EJB0106       Severity . . . . . :   00
Message type . . . . :   Information
Date sent  . . . . . :   11/06/02      Time sent  . . . . . :   13:37:57

Message . . . . :   WebSphere application server server1 ready.
Cause . . . . . :   WebSphere application server server1 in job
  023820/QEJBSVR/SERVER1 is ready to handle administrative requests on port
  9090.

                                                                       Bottom
Press Enter to continue.

F3=Exit   F6=Print   F9=Display message details   F12=Cancel
F21=Select assistance level
```

Figure 6-9 Port number used for the administrative console

8. Press F3 twice to exit.

Verifying if the MQ listener job is started

For every instance that you enabled for using embedded JMS, you will find a job named QEJBMQLSR (the MQ listener job) in the QEJBAS5 subsystem.

To determine which server the active QEJBMQLSR (MQ listener) job in subsystem QEJBAS5 belongs to, follow these steps:

1. Specify option **5** (Work with Job) on the option line next to the QEJBMQLSR job, and press Enter.
2. On the command line of the Work with Job display, type in **10** (Display job log, if active), and press Enter. You will get a screen similar to Figure 6-10.

```
Display Job Log
                                                          System:   RCHAS02B
Job . . :   QEJBMQLSR     User . . :   QEJBSVR       Number . . . :   020025

    Job 020025/QEJBSVR/QEJBMQLSR started on 12/14/02 at 15:42:52 in subsystem
      QEJBAS5 in QEJBAS5. Job entered system on 12/14/02 at 15:42:52.
    Printer device PRT01 not found. Output queue changed to QPRINT in library
      QGPL.
    WebSphere MQ job 6231 started for WAS_RCHAS07_arthur_arthur.
```

Figure 6-10 QEJBMQLSR job for instance arthur

3. The message in the joblog points to the corresponding application server.

6.5.5 Verifying the installation

The next logical step after you have started an application server is to verify that the server is operational. There are two ways to do this:

- Invoke the install verification script.
- Access one of the default sample Web applications.

Invoking the install verification script

To run the install verification script (`ivt`), follow these steps:

1. Start QShell, type the following on an iSeries command line and press Enter:

   ```
   STRQSH
   ```

2. Invoke the `ivt` script with the syntax:

   ```
   /QIBM/ProdData/WebAS5/Base/bin/ivt -instance instanceName
   ```

   Here, `instanceName` is the name of your WAS instance. For example:

   ```
   /QIBM/ProdData/WebAS5/Base/bin/ivt -instance webface
   ```

   The output produced by running this command is shown in Figure 6-11.

```
/QIBM/ProdData/WebAS5/Base/bin/ivt -instance webface
IVTL0095I: defaulting to host RCHAS02B.IBM.COM and port 10409
IVTL0010I: Connecting to the WebSphere Application Server RCHAS02B.IBM.COM on port:
10409
IVTL0015I: WebSphere Application Server RCHAS02B.IBM.COM is running on port: 10409


IVTL0050I: Servlet Engine Verification Status - Passed
IVTL0055I: JSP Verification Status - Passed
IVTL0060I: EJB Verification Status - Passed
IVTL0070I: IVT Verification Succeeded
IVTL0080I: Installation Verification is complete
$
```

Figure 6-11 Output ivt script for internal HTTP server - WEBFACE instance

Verifying installation using one of the sample Web applications

Three default sample applications are installed in each application server at create time (unless you specify `-nodefaultapps` with the `crtwasinst` script):

- HitCount
- snoop
- hello

You can use any of these applications for a quick verification test.

The HitCount application includes a Java servlet, a JavaServer Pages (JSP) file, and an enterprise bean version of the same logic.

To invoke these browser-based application examples, perform these steps:

1. Be sure that your WAS instance is already started and running; see “Verifying that the WAS environment has started” on page 119.

2. Open one of the following URLs in a Web browser (the URL is case-sensitive; the capitalization must be consistent), where `your.server.name` is the name of your iSeries server and `wasPort` is the internal HTTP port of your application server:

   a. For the HitCount sample, use the following URL:

      ```
      http://your.server.name:internalHTTPport/hitcount
      ```

      For example:

      ```
      http://RCHAS07:9080/hitcount
      ```

      The resulting page is similar to the one in Figure 6-12 on page 123.

   b. For the snoop servlet, which provides information about the client request as well as the snoop servlet itself, we type::

      ```
      http://your.server.name:internalHTTPport/snoop
      ```

      For our default application server, we type:

      ```
      http://RCHAS07:9080/snoop
      ```

      You should see a page similar to the one shown in Figure 6-12.

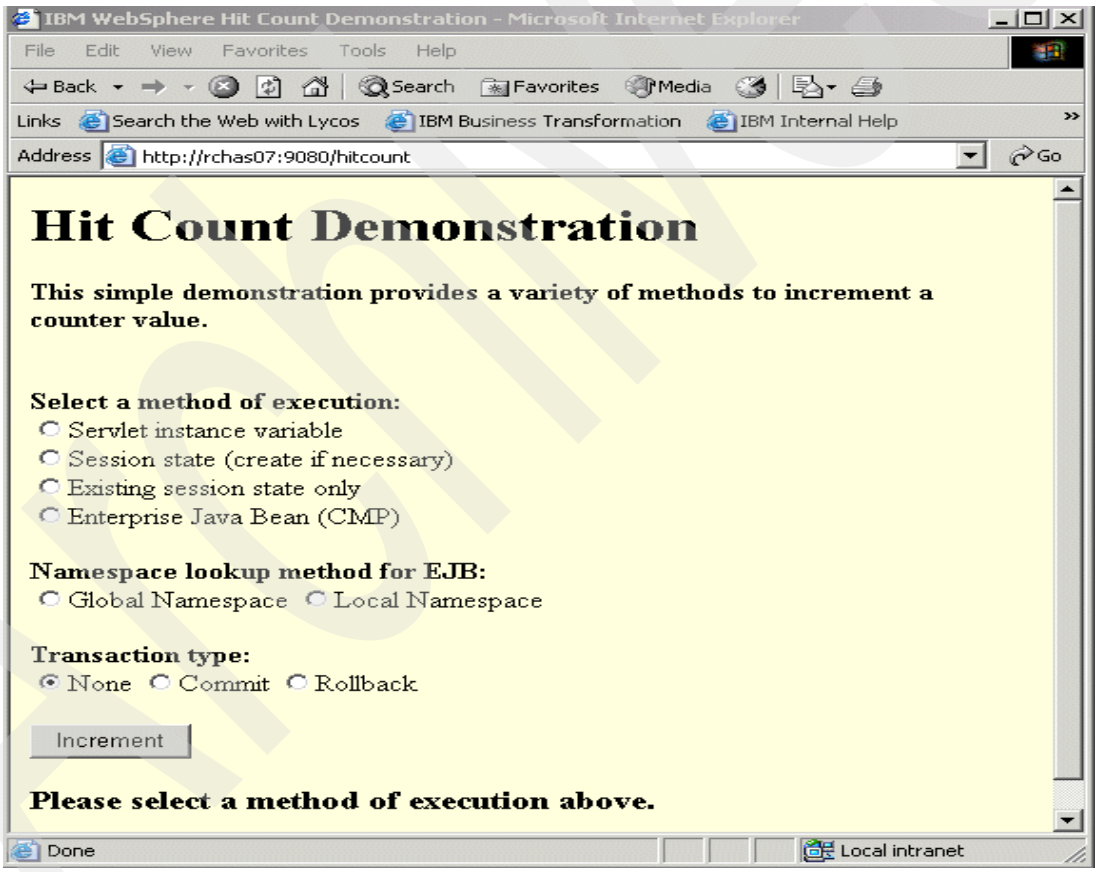


Figure 6-12 HitCount default application sample

Note: The HitCount sample works for the first 3 check boxes. The CMP bean option requires additional setup; see the online documentation via:

`http://publib.boulder.ibm.com/iseries/v1r1m0/websphere/ic2924/index.htm?info/rzaiz/50/was.htm`

c. For the hello sample, use the following URL:

```
http://your.server.name:internalHTTPport/hello
```

For example:

```
http://RCHAS07:9080/hello
```

d. For the snoop sample, use the following URL:

```
http://your.server.name:internalHTTPport/snoop
```

For example:

```
http://RCHAS07:9080/snoop
```

Note: If you use a non-default WAS instance, you may need to display the internal HTTP port of the application server you're using. The easiest way is to run the `dspwasinst` script. For example, if you want to display the port assignment for the *webface* server in the *webface* instance, you would run the following command:

```
dspwasinst -instance webface -server webface
```

6.5.6 Stopping an application server

You can stop an application server in one of three ways:

1. By running the `stopServer` script from a QShell command line.
2. By executing the end job (ENDJOB) command from an iSeries command line.
3. By executing the end subsystem (ENDSBS) command the QEJBAS5 subsystem. In this case all application servers on this system will be stopped.

The `stopServer` script

The `stopServer` script reads the configuration file for the specified server process and stops the server and dependent JMS listener job (QEJBMQLSR) if the server is enabled for JMS. This script is available in both WebSphere Application Server V5.0 for iSeries and WebSphere Application Server Network Deployment V5.0 for iSeries.

The `stopServer` script is located in the WAS_INSTALL_ROOT/bin directory where WAS_INSTALL_ROOT is located:

- `/QIBM/ProdData/WebAS5/Base/bin` for WebSphere Application Server V5.0 for iSeries
- `/QIBM/ProdData/WebAS5/ND/bin` for WebSphere Application Server Network Deployment V5.0 for iSeries

If you run the script from the Network Deployment directory, you stop a Network Deployment instance. This is a Deployment Manager instance, not an application server instance. For more information about Network Deployment, refer to Chapter 7, "WebSphere Application Server Network Deployment 5.0: configuration and administration" on page 263.

To run this script, your iSeries user profile must have *ALLOBJ authority.

For more information about syntax and available parameters for the `stopServer` script; see A.7.3, "Syntax and parameters for stopServer script" on page 533.

To stop an application server, follow these steps:

1. On the OS/400 command line, enter the STRQSH (Start QShell) command and press Enter.

2. On the QShell command line, use the **cd** command to change to the directory that contains the script:

   ```
   cd /QIBM/ProdData/WebAS5/Base/bin
   ```

3. Run the `stopServer` script:

   ```
   stopServer servername -instance instance_name
   ```

   The instance parameter is optional. If you omit it, the `stopServer` script will attempt to stop the specified server in the `default` instance.

   Consider these examples:

   – To stop the default application server (server1) for the default instance, execute the following command:

   ```
   stopServer server1
   ```

   – To stop the application server `webface` in the instance named `webface`, we type:

   ```
   stopServer webface -instance webface
   ```

   You should see messages similar to the ones shown in Figure 6-13.

```
> stopServer webface -instance webface
  ADMU0116I: Tool information is being logged in file
             /QIBM/UserData/WebAS5/Base/webface/logs/webface/stopServer.log
  ADMU3100I: Reading configuration for server: webface
  ADMU3201I: Server stop request issued. Waiting for stop status.
  ADMU4000I: Server webface stop completed.
  $
```

Figure 6-13 stopServer sample script

   – To stop the application server `david` in the instance named `team`, this is what we type (see Figure 6-14):

   ```
   stopServer david -instance team
   ```

```
stopServer david -instance team
ADMU0116I: Tool information is being logged in file
           /QIBM/UserData/WebAS5/Base/team/logs/david/stopServer.log
ADMU3100I: Reading configuration for server: david
ADMU3201I: Server stop request issued. Waiting for stop status.
ADMU4000I: Server david stop completed.
$
```

Figure 6-14 Stop server david in instance team

   – If you attempt to stop the server in a non-default instance, you see messages similar to those shown in Figure 6-15:

   ```
   stopServer diana
   ```

```
stopServer diana
ADMU0116I: Tool information is being logged in file
           /QIBM/UserData/WebAS5/Base/default/logs/diana/stopServer.log
ADMU3522E: No server by this name in the configuration: diana
$
```

Figure 6-15 Bad example for stopServer

Note: If the global security is enabled, you must use two additional parameters for the `stopServer` script: -username and -password. For example:

```
stopServer david -instance team -username team1 -password use1pwd
```

Ending an application server job via OS/400 command ENDJOB

You can stop an application server by running the ENDJOB OS/400 command against your application server job. Follow these steps:

1. Type `WRKACTJOB` on the OS/400 command line and press Enter.
2. Find your application server job in the QEJBAS5 subsystem.
3. Position the cursor on the options line of your application server job and type **4** (this is the End Job option) as shown in Figure 6-16.

```
                          Work with Active Jobs                    RCHAS02B
                                                              12/16/02  01:10:20
CPU %:     .8     Elapsed time:   00:31:43     Active jobs:   253

Type options, press Enter.
  2=Change   3=Hold   4=End   5=Work with   6=Release   7=Display message
  8=Work with spooled files   13=Disconnect ...

Opt  Subsystem/Job  User        Number  Type  CPU %  Threads
     QEJBAS5        QSYS        021405  SBS      .0        1
       ARTHUR       QEJBSVR     021408  ASJ      .0       21
       QEJBMQLSR    QEJBSVR     021442  BCI      .0        3
       QEJBMQLSR    QEJBSVR     021445  BCI      .0        3
       SERVER1      QEJBSVR     021406  ASJ      .0       22
 4     USCHI        QEJBSVR     021410  ASJ      .0       19
       WEBFACE      QEJBSVR     021407  ASJ      .0       19
     QHTTPSVR       QSYS        013877  SBS      .0        1
       ADMIN        QTMHHTTP    013878  BCH      .0        1
                                                                    More...
Parameters or command
===>
F3=Exit   F5=Refresh   F7=Find   F10=Restart statistics   F11=Display status
F12=Cancel   F17=Top   F18=Bottom   F23=More options      F24=More keys
```

Figure 6-16 End the application server uschi via ENDJOB from WRKACTJOB OS/400 display

4. If you need to change any of the parameters for the ENDJOB command, press `F4` (see Figure 6-17).

Note: Set the DELAY parameter to an adequate amount of time to stop the application server in a controlled manner.

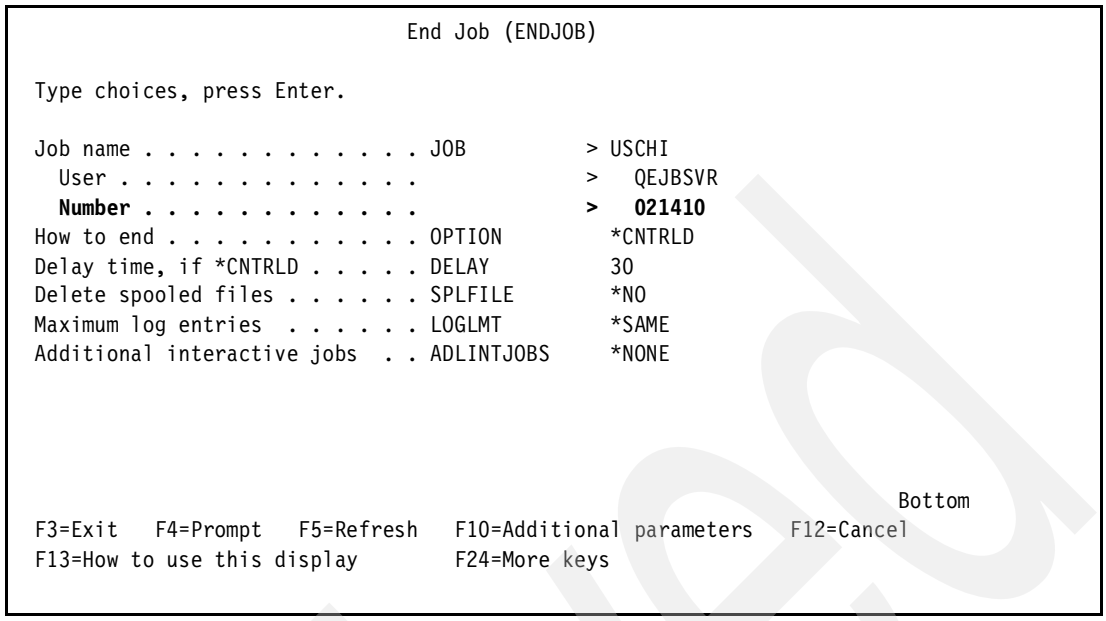


Figure 6-17 End the application server diana via ENDJOB command

5. Press Enter. The ENDJOB command is submitted.

Note: When you use the ENDJOB command to stop an application server the depending JMS listener job QEJBMQLSR is not stopped!!! You have to stop this job also. In order to find the corresponding JMS listener job for the application server; see “Verifying if the MQ listener job is started” on page 121.

6. On some systems the application server job may show the status of THDW for several minutes. This is due to the JVM cleaning up allocated memory and other resources.

Important: Always use the `*CNTRLD` value for the `OPTION` parameter, allowing the application server job to shut down in a controlled manner.

However, in general, use the `stopServer` script as the preferred way of stopping an application server.

If you try to run ENDJOB command against a server with the global security enabled, you must provide the user ID and corresponding password in the file `<instnace_directory>/properties/soap.client.props`.

Stopping the WebSphere Application Server environment

You may choose to use the ENDSBS command to stop all application server jobs on the system. In all cases, it is important to end the job gracefully so that the job can finish any tasks that are currently in progress, clean up any open connections, and end multiple threads in an appropriate order. That’s why you should always use the ENDSBS command in a controlled manner.

You can stop (end) the WebSphere Application Server environment by invoking the End Subsystem (ENDSBS) command:

```
ENDSBS SBS(QEJBAS5) OPTION(*CNTRLD) DELAY(600)
```

Here, the DELAY parameter specifies the adequate number of seconds for the WebSphere subsystem to end.

This command ends all jobs running in the QEJBAS5 subsystem. Additionally, if you are using the embedded JMS service, any WebSphere MQ jobs which were started in the QMQM subsystem on your behalf are ended.

*Using the *IMMED value for the OPTION parameter*

In OS/400 release V5R1 and higher, new support has been added to call the job termination signal handler for a job (if one is enabled) when you run ENDSBS OPTION(*IMMED).

If there are conditions under which you must run ENDSBS OPTION(*IMMED), you should create a data area which will specify the amount of time (in seconds) available for handling the job termination signal. 120 seconds should be adequate:

```
CRTDTAARA DTAARA(QSYS/QENDJOBLMT) TYPE(*DEC) LEN(5 0) VALUE(120) TEXT('TIME LIMIT IN
SECONDS')
```

In order to verify if the delay time is sufficient for an application server job to shut down gracefully, look at the job log. One of the last messages (message ID is EJB0107) in the job log should display the number of seconds that it took to end the application server in a controlled manner.

If no message is found in the job log, it is a good indication that you need to increase the amount of time specified in the QENDJOBLMT data area. Adjust this value using the Change Data Area (CHGDTAARA) command:

```
CHGDTAARA DTAARA(QSYS/QENDJOBLMT) VALUE(600)
```

The maximum value allowed is 3600. The minimum value allowed is 30.

6.5.7 Changing a WAS application server via the script

The chgwassvr script allows you to change properties of an application server. You can change the port numbers for the application server, for example, or enable or disable the embedded JMS. This script is available in both WebSphere Application Server V5.0 for iSeries and WebSphere Application Server Network Deployment V5.0 for iSeries.

The chgwassvr script is located in the WAS_INSTALL_ROOT/bin directory where WAS_INSTALL_ROOT is located:

- /QIBM/ProdData/WebAS5/Base/bin for WebSphere Application Server V5.0 for iSeries
- /QIBM/ProdData/WebAS5/ND/bin for WebSphere Application Server Network Deployment V5.0 for iSeries

In this section we show how to use this script for WebSphere Application Server V5.0 for iSeries. If you use the script from the Network Deployment directory, refer to Chapter 7, “WebSphere Application Server Network Deployment 5.0: configuration and administration” on page 263.

For information about syntax and available parameters for the chgwassvr script; see A.7.4, “Syntax and parameters for chgwassvr script” on page 534.

To change the configuration of a WAS instance with the chgwassvr script, follow these steps:

1. On the OS/400 command line, enter the STRQSH (Start QShell) command and press Enter.

2. On the QShell command line, use the **cd** command to change to the directory that contains the script, in our case the WebSphere Application Server:

   ```
   cd /QIBM/ProdData/WebAS5/Base/bin
   ```

3. Type in the `chgwassrv` command following the syntax and using the parameters for the elements you want to change. We provide several examples:

   – To enable the embedded JMS for the default instance we type:

     ```
     chgwassvr -server server1 -instance default -embeddedjms yes
     ```

     You will see messages similar to those shown in Figure 6-18.

     The MQ listener job QEJBMQLSR is not started after `chgwassrv` is done; you have to stop and start the application server to make this change effective.

```
chgwassvr -server server1 -instance default -embeddedjms yes
ADCP0005I: Using cell RCHAS02B, node RCHAS02B and server server1.
ADCP0006I: Embedded JMS enabled.
$
```

Figure 6-18 Enable default server for JMS

   – To disable JMS for the david server in the team instance, run the following command:

     ```
     chgwassvr -server david -instance team -embeddedjms no
     ```

     You will see messages similar to those shown in Figure 6-19.

     The MQ listener job QEJBMQLSR is ended after the `chgwassrv` script completes.

```
> chgwassvr -server david -instance team -embeddedjms no
  ADCP0005I: Using cell RCHAS02B_team, node RCHAS02B_team and server david.
  ADCP0007I: Embedded JMS disabled.
  $
```

Figure 6-19 Disable JMS for server david

   – You can change the internal HTTP port for server `webface` in the webface instance by running the `chgwassvr` script with the following parameter:

     ```
     chgwassvr -server webface -instance webface -inthttp 10420
     ```

     You will see messages similar to those shown in Figure 6-20.

```
chgwassvr  -server webface  -instance webface  -inthttp 10420
ADCP0005I: Using cell RCHAS02B_webface, node RCHAS02B_webface and server webface.
ADCP0001I: Ports changed successfully.
ADCP0002I: The following ports were changed to the specified value:
             Internal HTTP port:   10,420
$
```

Figure 6-20 Change internal HTTP port with chgwassvr script

6.5.8 Displaying a WAS instance via the dspwasinst script

To display the most important configuration details of a WAS instance, use the `dspwasinst` script. This script is available in both WebSphere Application Server V5.0 for iSeries and WebSphere Application Server Network Deployment V5.0 for iSeries.

The dspwasinst script is located in the WAS_INSTALL_ROOT/bin directory where WAS_INSTALL_ROOT is located:

- /QIBM/ProdData/WebAS5/Base/bin for WebSphere Application Server V5.0 for iSeries
- /QIBM/ProdData/WebAS5/ND/bin for WebSphere Application Server Network Deployment V5.0 for iSeries

If you use the script from the Network Deployment directory, you display a Network Deployment instance. This is a Deployment Manager instance, not an application server instance. For more information about Network Deployment refer to Chapter 7, "WebSphere Application Server Network Deployment 5.0: configuration and administration" on page 263.

To display the configuration of a WAS instance with the dspwasinst script, follow these steps:

1. On the OS/400 command line, enter the STRQSH (Start QShell) command and press Enter.
2. On the QShell command line, use the **cd** command to change to the directory that contains the script, in our case:

   ```
   cd /QIBM/ProdData/WebAS5/Base/bin
   ```

3. Run the dspwasinst script with a single parameter - instance:

   ```
   dspwasinst -instance webface
   ```

   Figure 6-21 shows the output of the dspwasinst script.

```
dspwasinst -instance webface
ADCP0005I: Using cell RCHAS02B_webface, node RCHAS02B_webface and server *ALL.
Display WAS instance:
  Instance name: webface
  Instance type: Base Application Server
  Cell: RCHAS02B_webface
  Node: RCHAS02B_webface

  Information for server: webface
    Installed applications:
      DefaultApplication
      ivtApp
      adminconsole
      WebFacing1

     Ports in use:
          10410   Administrative console port
          10411   Administrative console SSL-enabled port
          10400   Name service port
          10405   Soap port
          10404   Data replication service client port
          10401   Internal Java Message Service server port
          10402   Queued Java Message Service server port
          10403   Direct Java Message Service server port
          10406   SAS SSL server authentication listener port
          10408   CSIV2 server authentication listener port
          10407   CSIV2 mutual authentication listener port
$
```

Figure 6-21 dspwasinst instance webface

6.5.9 Deleting a WebSphere Application Server instance

To delete a instance, run the `dltwasinst` script from a QShell command line. The `dltwasinst` script removes an instance and the files associated with it. It also deletes any embedded JMS brokers that are associated with the instance. This script is available in both WebSphere Application Server V5.0 for iSeries andWebSphere Application Server Network Deployment V5.0 for iSeries.

The `dltwasinst` script is located in the WAS_INSTALL_ROOT/bin directory where WAS_INSTALL_ROOT is located:

- `/QIBM/ProdData/WebAS5/Base/bin` forWebSphere Application Server V5.0 for iSeries
- `/QIBM/ProdData/WebAS5/ND/bin` forWebSphere Application Server Network Deployment V5.0 for iSeries

If you use the script from the Network Deployment directory, you delete a new Network Deployment instance, this is a Deployment Manager instance, not an application server instance. For more information about Network Deployment, refer to Chapter 7, “WebSphere Application Server Network Deployment 5.0: configuration and administration” on page 263.

To run this script, your iSeries user profile must have *ALLOBJ authority. Before deleting any instance, make sure the instance is not active.

Perform the following steps to delete an instance:

1. On the OS/400 command line, run the STRQSH (Start QShell) command.
2. On the QShell command line, use the **cd** command to change to the directory that contains the script. For example:

   For WebSphere Application Server:

   ```
   cd /QIBM/ProdData/WebAS5/Base/bin
   ```

   Or for WebSphere Application Server Network Deployment

   ```
   cd /QIBM/ProdData/WebAS5/ND/bin
   ```

3. Run the `dltwasinst` script:

   ```
   dltwasinst -instance instance
   ```

   Here, instance is the name of the instance you want to delete.

   For example:

   ```
   dltwasinst -instance arthur
   ```

   Because the arthur instance is JMS enabled, you will see similar message as shown in Figure 6-22.

```
dltwasinst -instance arthur
Checking if any servers are running...
Checking if MQ managers are running...
Deleting MQ Manager for server arthur...
Deleting instance arthur...
$
```

Figure 6-22 Delete instance arthur

6.6 Configuring an HTTP server instance

An HTTP server is required to support requests for servlets and JavaServer Pages resources managed by WebSphere Application Server. For a production environment, an external HTTP server is recommended. It is also needed if you plan to use Secure Sockets Layer (SSL) protocol.

There are two HTTP servers on iSeries that are available to work as an external HTTP server together with WebSphere Application Server on iSeries Version 5.0:

- IBM HTTP server for iSeries (powered by apache)
- Lotus Domino Web server

Note: The IBM HTTP Server for iSeries (original) is not supported for WebSphere Application Server V5.0.

It is also possible to configure a remote HTTP Servers residing on other supported platforms which route requests to WebSphere Application Server running on the iSeries server. For more information, see:

http://publib.boulder.ibm.com/iseries/v1r1m0/websphere/ic2924/index.htm?info/rzaiz/50/admin/rmthttp.htm

We now describe the creation of an HTTP server instance on iSeries and the necessary configuration steps that enable this instance to communicate with WebSphere Application Server Version 5.0 on an iSeries server. The descriptions are organized as follows:

1. HTTP server for iSeries (powered by apache), general steps, starting at “Configuring IBM HTTP Server (powered by Apache) for iSeries” on page 132:
   - On iSeries with OS/400 V5R1, starting at “Configuring IBM HTTP Server (powered by Apache) in OS/400 V5R1” on page 133.
   - On iSeries with OS/400 V5R2, starting at “Configuring IBM HTTP Server (powered by Apache) in OS/400 V5R2” on page 143.
2. Lotus Domino Web server, starting at “Configuring Lotus Domino Web server” on page 157.

Note: If you already have an existing HTTP instance on iSeries, that you want to use with the WebSphere Application Server, do only the configuration steps for WebSphere Application Server in this HTTP instance:

- For iSeries server with OS/400 V5R1, do the steps beginning with step 19 on page 142.
- For iSeries server with OS/400 V5R2, do the steps beginning with step 19 on page 155.

6.6.1 Configuring IBM HTTP Server (powered by Apache) for iSeries

All instances of the IBM HTTP Server for iSeries are running in the QHTTPSVR subsystem, and each HTTP server instance starts multiple jobs. Whenever you change your HTTP server instance configuration, you must stop and then start your HTTP server instance.

Before you can create or configure an IBM HTTP Server (powered by Apache), ensure that the administration instance of IBM HTTP server is started.

Starting the *ADMIN instance of IBM HTTP Server

Perform these steps to start the administration HTTP server instance:

1. From the OS/400 command line, type the following command and press Enter:

   STRTCPSVR SERVER(*HTTP) HTTPSVR(*ADMIN)

2. Verify that the administration HTTP server instance is running. Use the Work with Active Jobs (WRKACTJOB) command from the iSeries command line:

   WRKACTJOB SBS(QHTTPSVR)

3. If the administration HTTP server is started, you will see a screen similar to the one shown in Figure 6-23.

```
                          Work with Active Jobs                    AS07
                                                        11/10/02  12:25:13
CPU %:     .6     Elapsed time:   00:00:37     Active jobs:   204

Type options, press Enter.
  2=Change   3=Hold   4=End   5=Work with   6=Release   7=Display message
  8=Work with spooled files   13=Disconnect ...

Opt  Subsystem/Job  User        Type  CPU %  Function        Status
     QHTTPSVR       QSYS        SBS      .0                 DEQW
       ADMIN        QTMHHTTP    BCH      .0  PGM-QZHBHTTP   SIGW
       ADMIN        QTMHHTTP    BCI      .0  PGM-QZSRLOG    SIGW
       ADMIN        QTMHHTTP    BCI      .0  PGM-QZSRHTTP   SIGW
       ADMIN        QTMHHTTP    BCI      .0  PGM-QYUNLANG   TIMW
       ADMIN        QTMHHTTP    BCI      .0  PGM-QYUNLANG   TIMW




                                                                Bottom
Parameters or command
===>
F3=Exit   F5=Refresh       F7=Find      F10=Restart statistics
F11=Display elapsed data   F12=Cancel   F23=More options   F24=More keys
```

Figure 6-23 Active HTTP server administration instance

6.6.2 Configuring IBM HTTP Server (powered by Apache) in OS/400 V5R1

To configure a new instance of IBM HTTP Server (powered by Apache) to communicate with WebSphere Application Server on an iSeries system with OS/400 V5R1, use the basic configuration wizard, which provides the necessary IBM HTTP Server for iSeries Configuration and Administration forms.

For every WAS instance, you have to create a separate HTTP server instance. Follow these steps:

1. Using a browser, open the IBM HTTP Server Configuration and Administration page:
   - Enter the following URL in your browser:

     http://<server name>:2001/

     Here, <server name> is the TCP host/domain name of your iSeries server; see 2.9, “Starting, configuring, and verifying TCP/IP” on page 21.

     In our case, we type the URL:

     http://RCHAS07:2001

2. Enter a valid user ID and password for your iSeries in the pop-up window and click **OK**.

3. At the AS/400 Tasks page, select **IBM HTTP Server for AS/400** (see Figure 6-24).

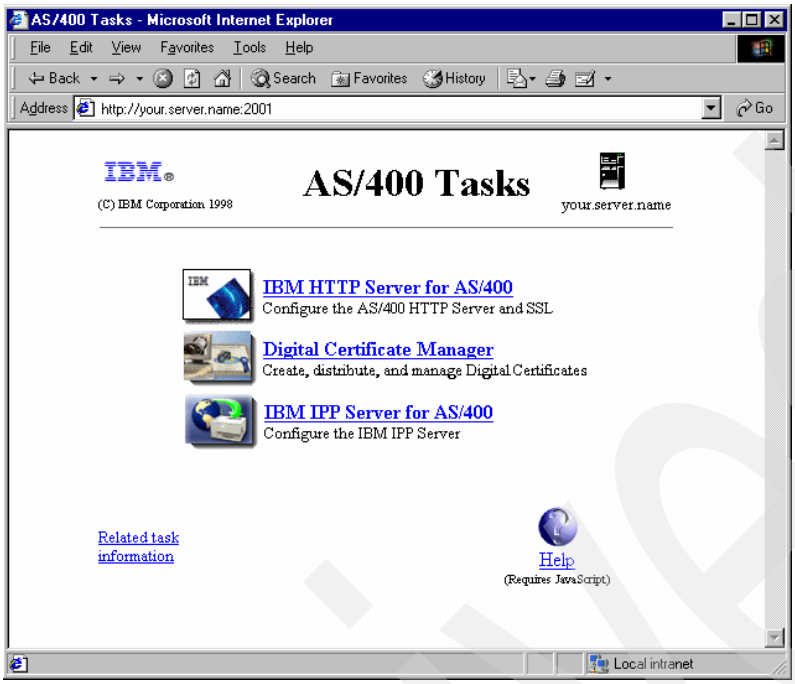


Figure 6-24 V5R1 AS/400 Tasks

4. The IBM HTTP Server for iSeries page is displayed. In the left frame, click **Configuration and Administration** (see Figure 6-25).

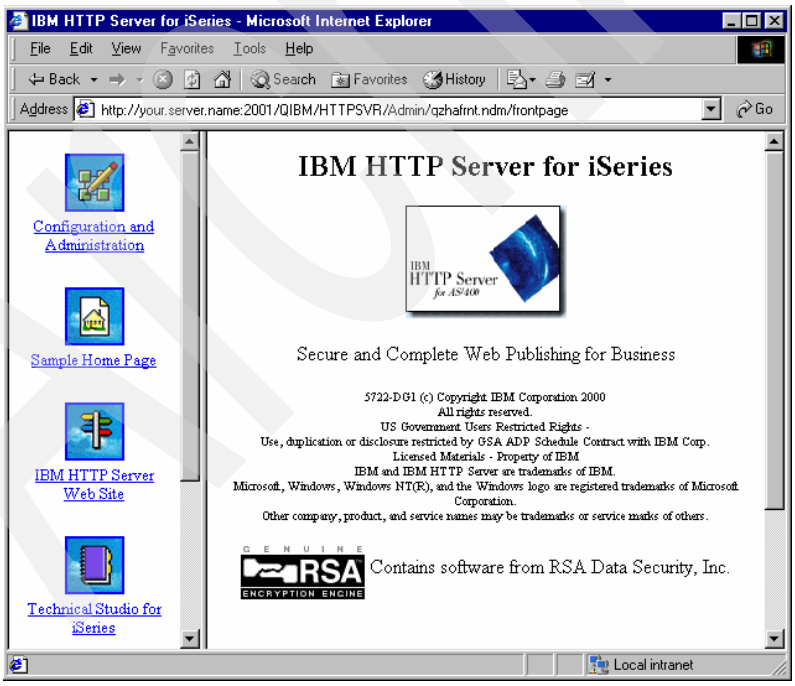


Figure 6-25 OS/400 V5R1 Configuration and Administration

5. The Administration page is displayed. To start the Create HTTP Server wizard, click **Create HTTP Server** in the navigation bar on the left (see Figure 6-26).

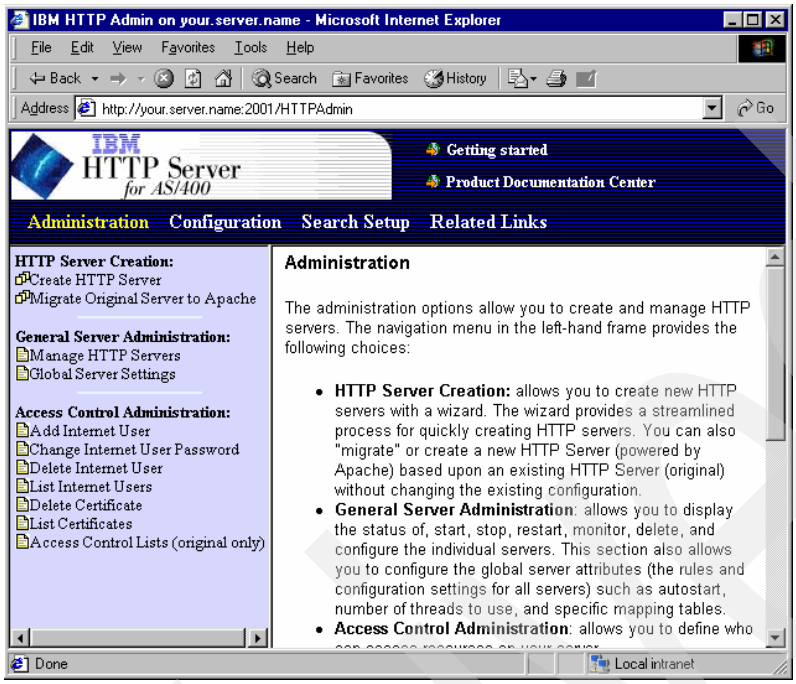


Figure 6-26 OS/400 V5R1 Create HTTP Apache server

6. Select **HTTP server** (**powered by Apache**) and then click **Next (**see Figure 6-27).

Note: WebSphere Application Server Version 5.0 for iSeries is only supported with the IBM HTTP Server for iSeries (powered by Apache) or Lotus Domino for iSeries HTTP server. The IBM HTTP Server for iSeries original is **not** supported.

Figure 6-27 OS/400 V5R1 Create HTTP Server (powered by Apache)

7. Enter a name for the HTTP server instance and click **Next** (see Figure 6-28). This name will appear in the QHTTPSVR subsystem as the job name, when the HTTP server is running.

Note: We used the same name for the HTTP server instance as for our WAS instances.

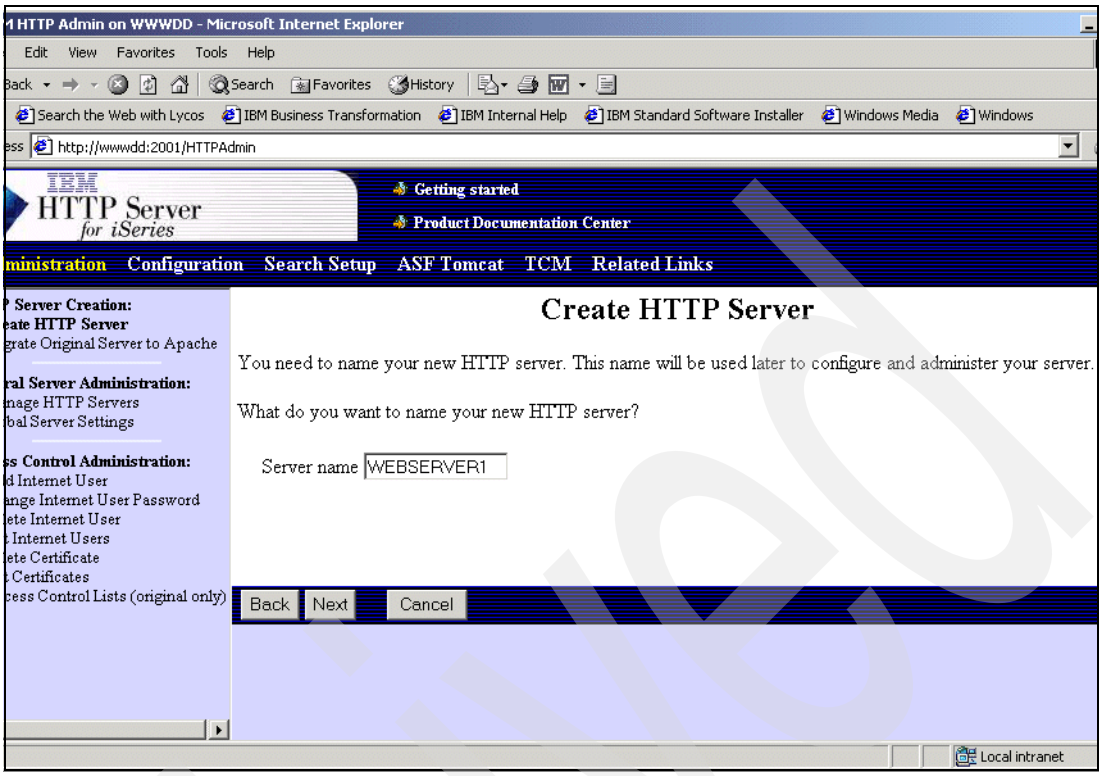


Figure 6-28 OS/400 V5R1 Set Http server name

8. Select **No** for the question regarding creating your new HTTP Server based on an existing one, and then click **Next** (see Figure 6-29).

Figure 6-29 OS/400 V5R1 No creation from existing Original HTTP server

9. In the next panel (see Figure 6-30), enter a name for the server root directory.

   You can specify any name for the server root or work with the default root name which will be already filled in. The directory you specify here will be created in the Integrated File System (IFS) of the iSeries server, if it doesn't exists.

10. Click **Next**.

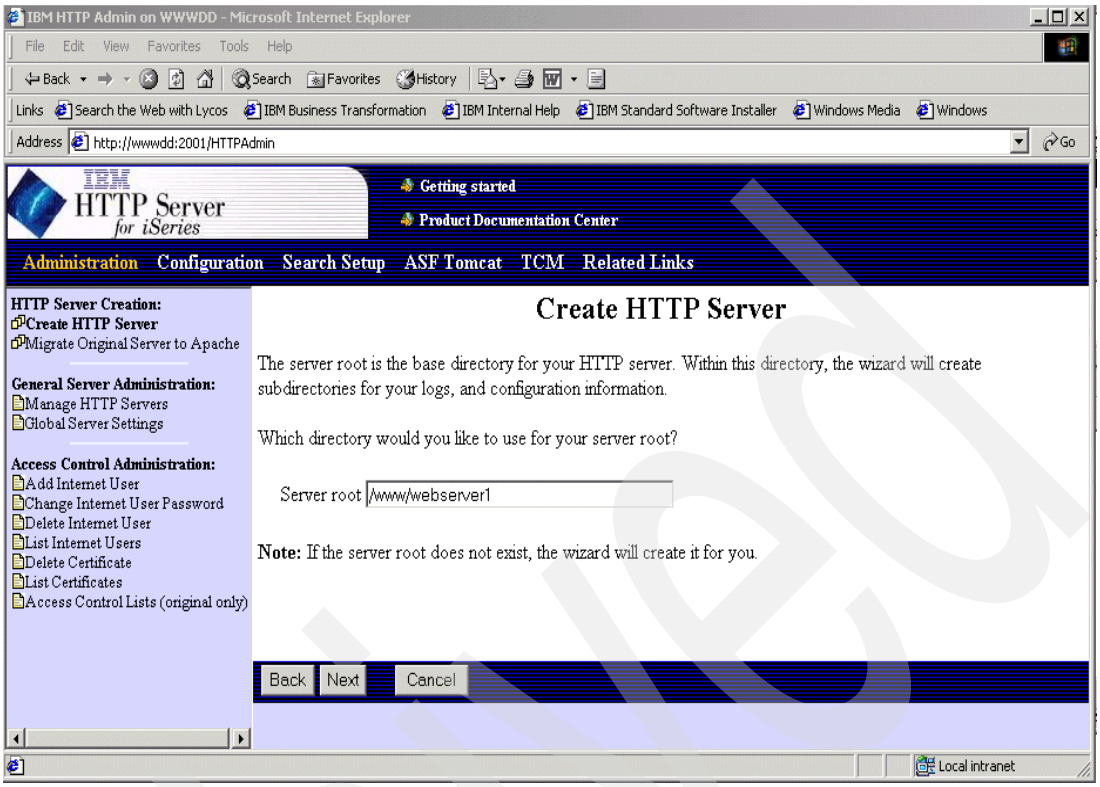


Figure 6-30 OS/400 V5R1 Set server root

11. In the next panel (see Figure 6-31), enter the name for the document root.
12. Click **Next**.

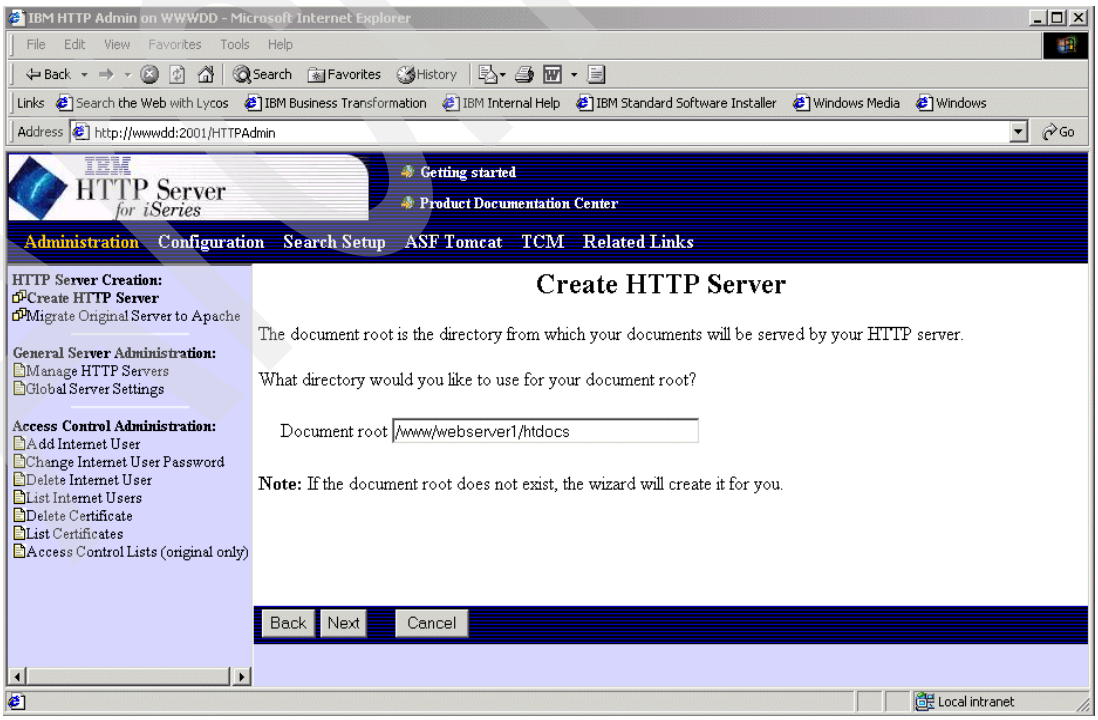


Figure 6-31 OS/400 V5R1 Set document root

13. Select the IP address and port (see Figure 6-32):
    - In the IP address field, select **All addresses**.

      You can also select a dedicated TCP/IP address to process your requests.
    - The default port number is 80.

      The port number you specify here has to match the external HTTP port number you use for your specific WAS instance. For example, the port number you specify with `-exthttp` parameter for the `crtwasinst` script (see “Creating a new WAS instance” on page 113).
14. Click **Next**.

Figure 6-32 OS/400 V5R1 Select all addresses

15. Select a logging option (see Figure 6-33). We recommend that you select the **Combined log file** option.

    The logs are stored in the /<*server_root*>/logs directory, where <*server_root*> is the server root that you set in step 9 on page 138.
16. Click **Next**.

Figure 6-33 OS/400 V5R1 Combined logs

17. Review your settings which are summarized in the next screen (see Figure 6-34).
    - To create the HTTP server instance, click **Finish**.
    - To make changes, click **Back**.

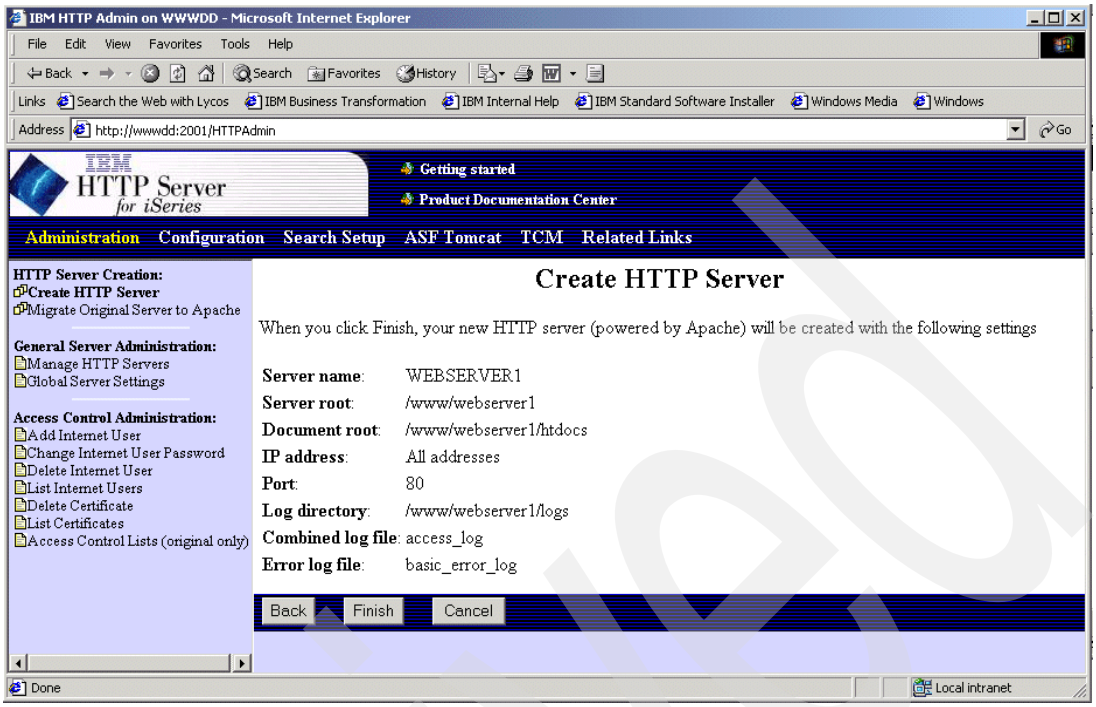


Figure 6-34 OS/400 V5R1 Configuration summary

18. The next panel acknowledges the creation of the new HTTP server.
19. After you created the HTTP server instance, configure the instance so that it can communicate with the WebSphere Application Server.
    - Click **Configure** (see Figure 6-35).

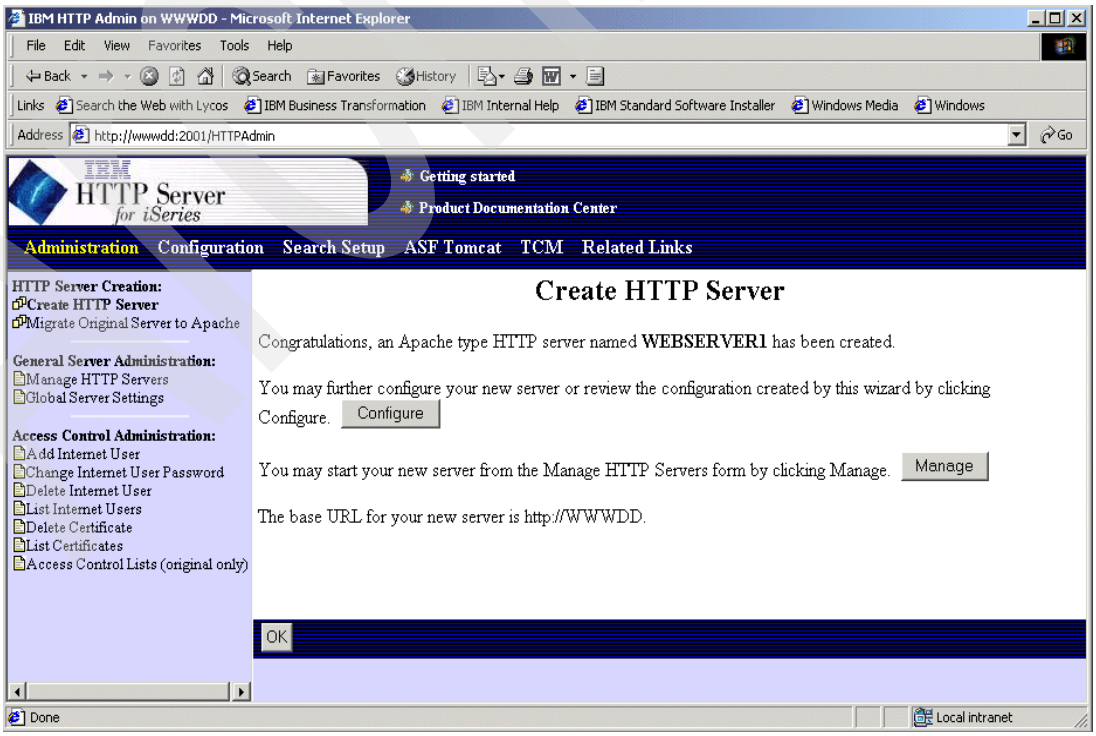


Figure 6-35 OS/400 V5R1 Configuration acknowledgement

– Under the Dynamic Content heading, select **WebSphere Application Server** (see Figure 6-36).

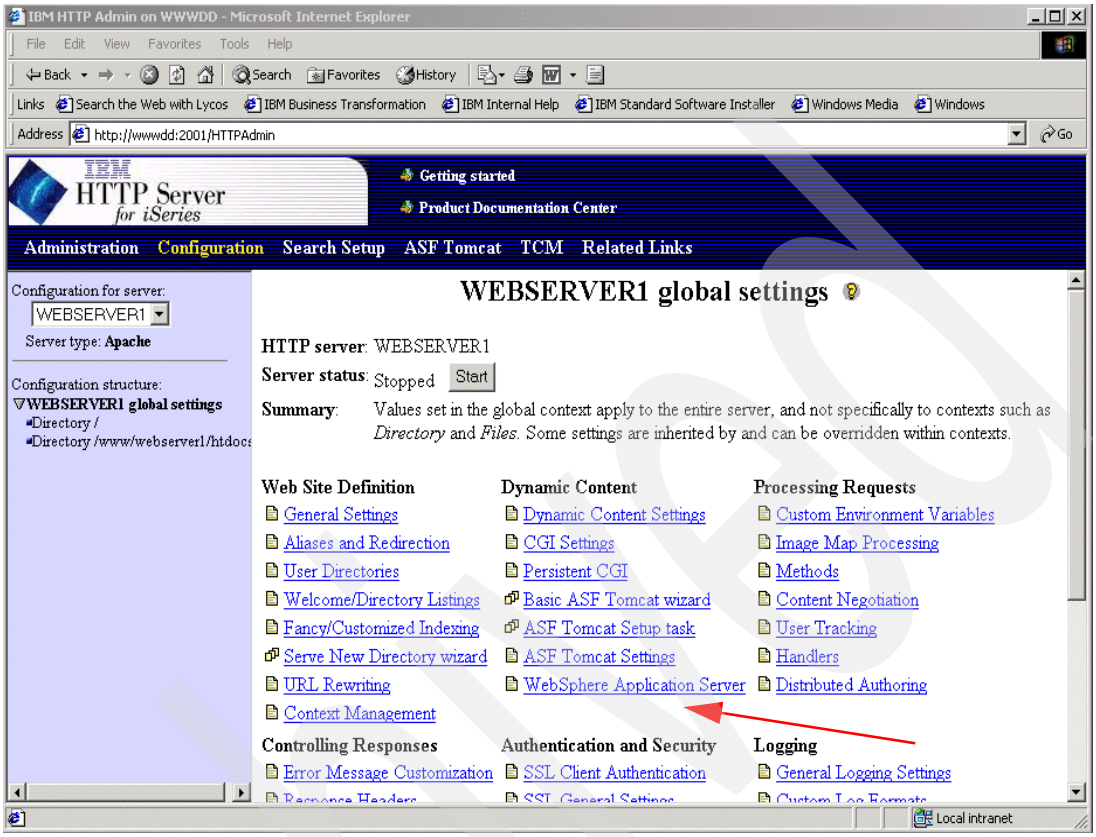


Figure 6-36 OS/400 V5R1 Configure HTTP server

20. The selections available on this page depend on which versions of WebSphere are currently installed on the iSeries server.

    Select **WebSphere version 5** and choose the desired WAS instance from the selection list, and click **Apply**.

21. Start your HTTP server instance as described in “Starting IBM HTTP Server for iSeries” on page 176.

6.6.3 Configuring IBM HTTP Server (powered by Apache) in OS/400 V5R2

To configure a new instance of IBM HTTP Server (powered by Apache) to communicate with WebSphere Application Server on an iSeries system with OS/400 V5R2, use the basic configuration wizard, which provides the necessary IBM HTTP Server for iSeries Configuration and Administration forms.

For every WAS instance, you have to create a separate HTTP server instance. Follow these steps:

1. Using a browser, open the IBM HTTP Server Configuration and Administration page:
   - Enter the following URL in your browser:

     http://<server name>:2001/

     Here, <server name> is the TCI/IP host/domain name of your iSeries server; see 2.9, “Starting, configuring, and verifying TCP/IP” on page 21.
   - In our case, we type the URL:

     http://RCHAS07:2001
2. In the pop-up window, enter a valid user ID and password for your iSeries server.
3. At the iSeries Tasks page, select **IBM HTTP Server for iSeries** (see Figure 6-37).

Figure 6-37 OS/400 V5R2 iSeries Tasks

4. On the SETUP tab in the HTTP Server for iSeries page, **expand Task and Wizards** in the left hand frame and select **Create New HTTP Server** (see Figure 6-38).

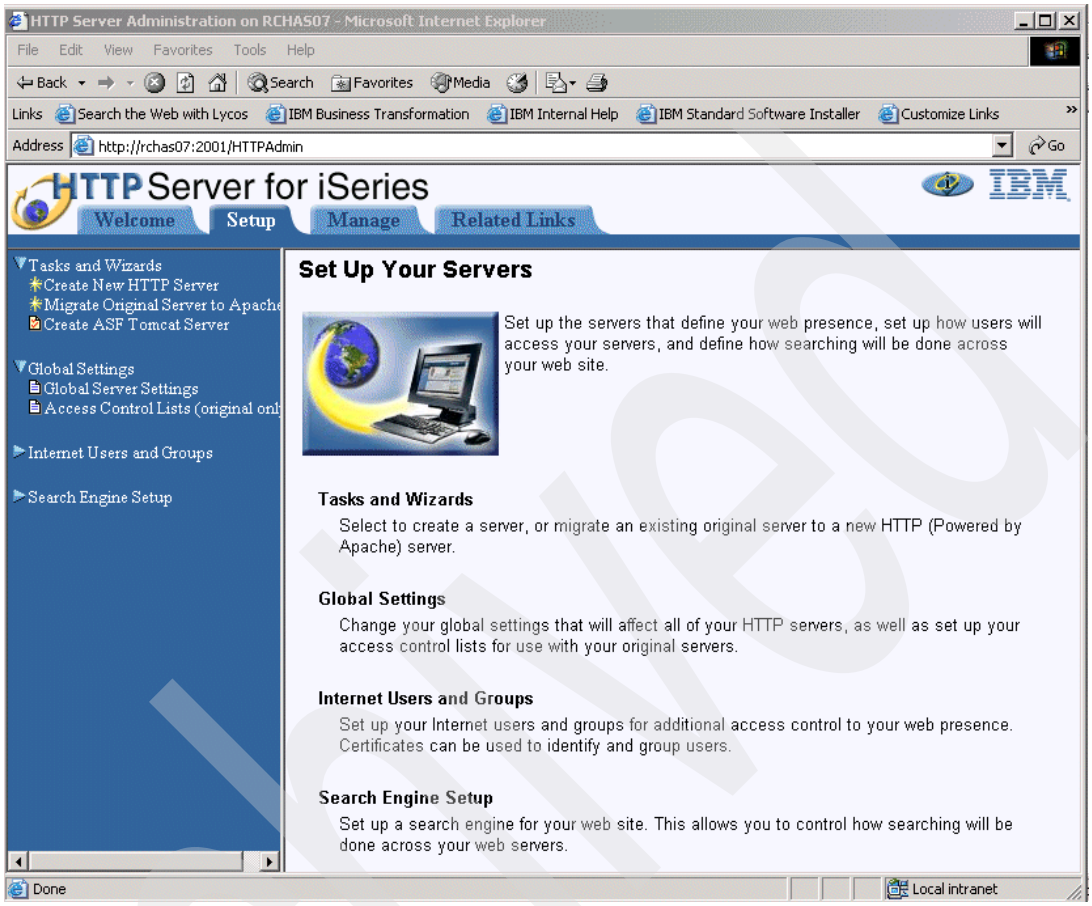


Figure 6-38 OS/400 V5R2 Setup HTTP Server for iSeries

5.Select **HTTP Server (powered by Apache)** and click **Next**; see Figure 6-39.

Note: WebSphere Application Server Version 5.0 for iSeries is only supported with the IBM HTTP Server for iSeries (powered by Apache) or Lotus Domino for iSeries HTTP server. The IBM HTTP Server for iSeries original is **not** supported.

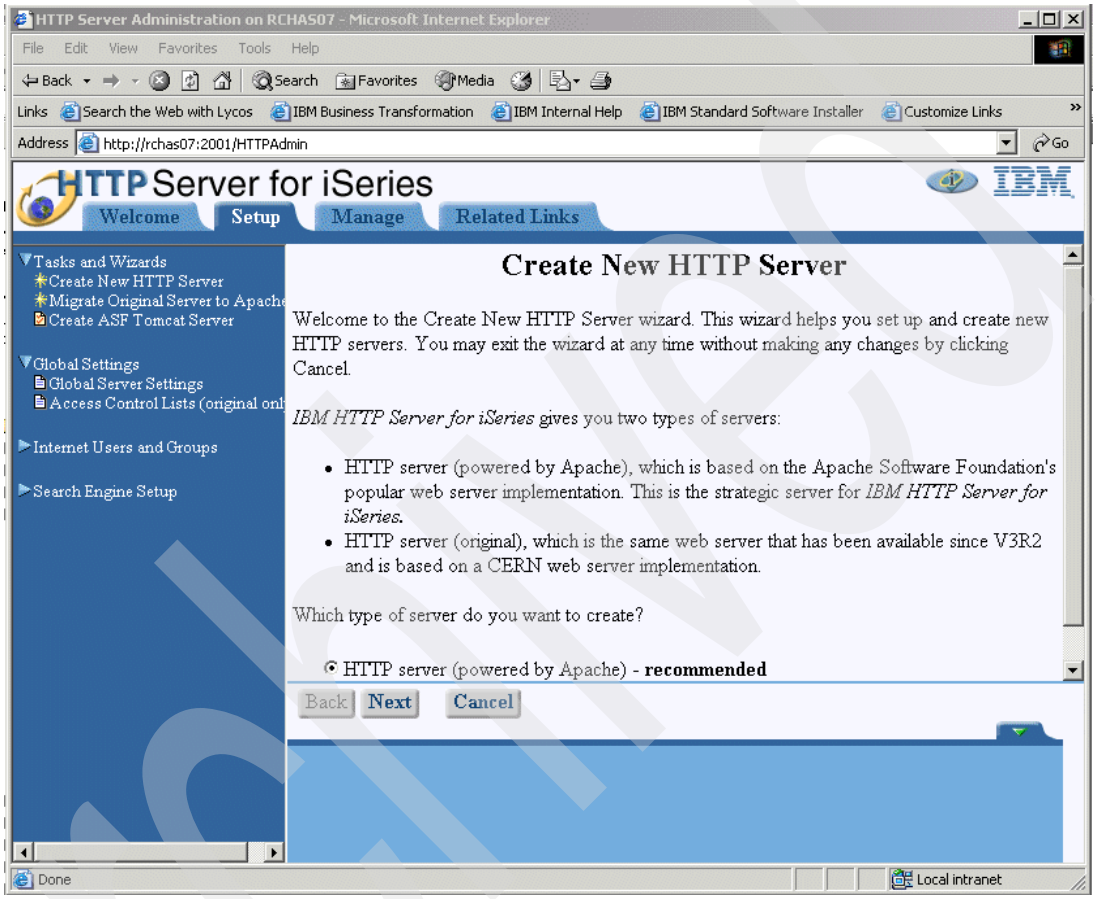


Figure 6-39 OS/400 V5R2 Create HTTP server (powered by Apache)

6. Enter the name for the HTTP server instance and click **Next** (see Figure 6-40). This name will appear in the QHTTPSVR subsystem as job name, when the HTTP server is running.

Note: We used the same name for the HTTP server instance as for our WAS instances.

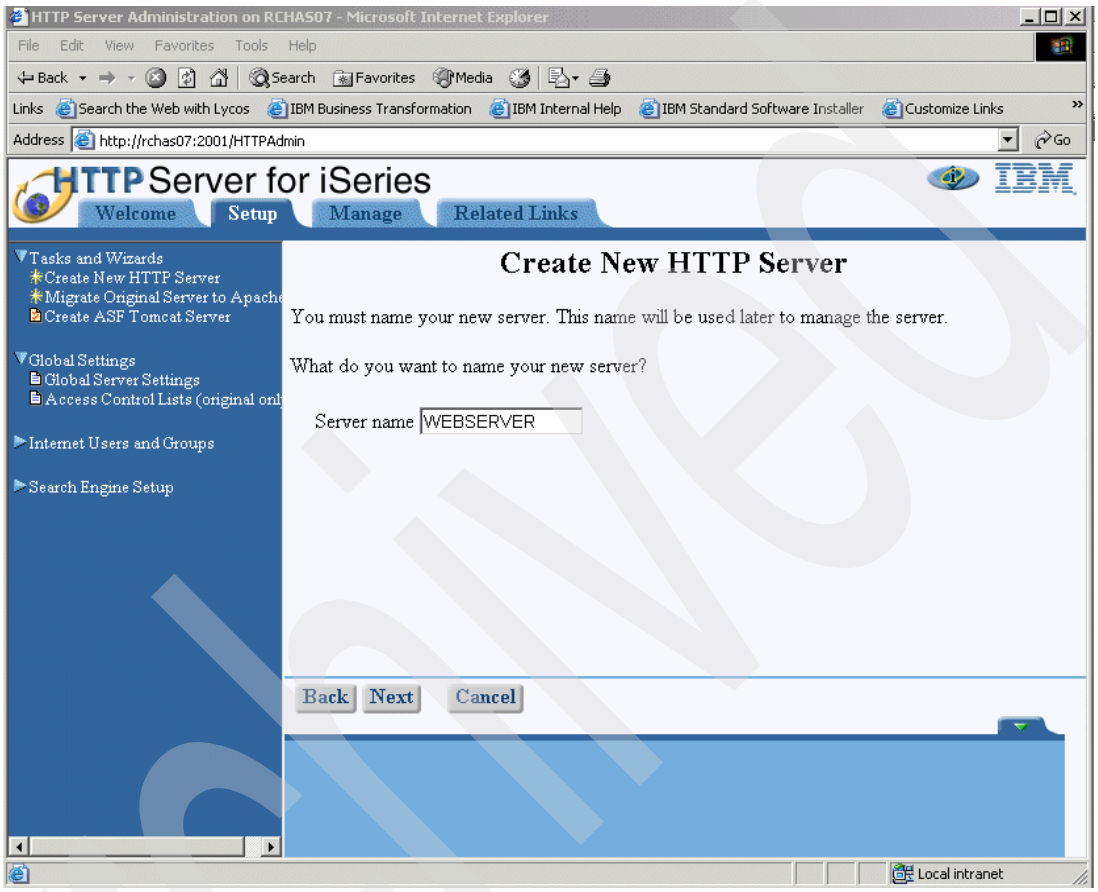


Figure 6-40 OS/400 V5R2 Set HTTP Server Name

7. Select **No** for the question regarding creating your new HTTP Server based on an existing one, and then click **Next (**see Figure 6-41).

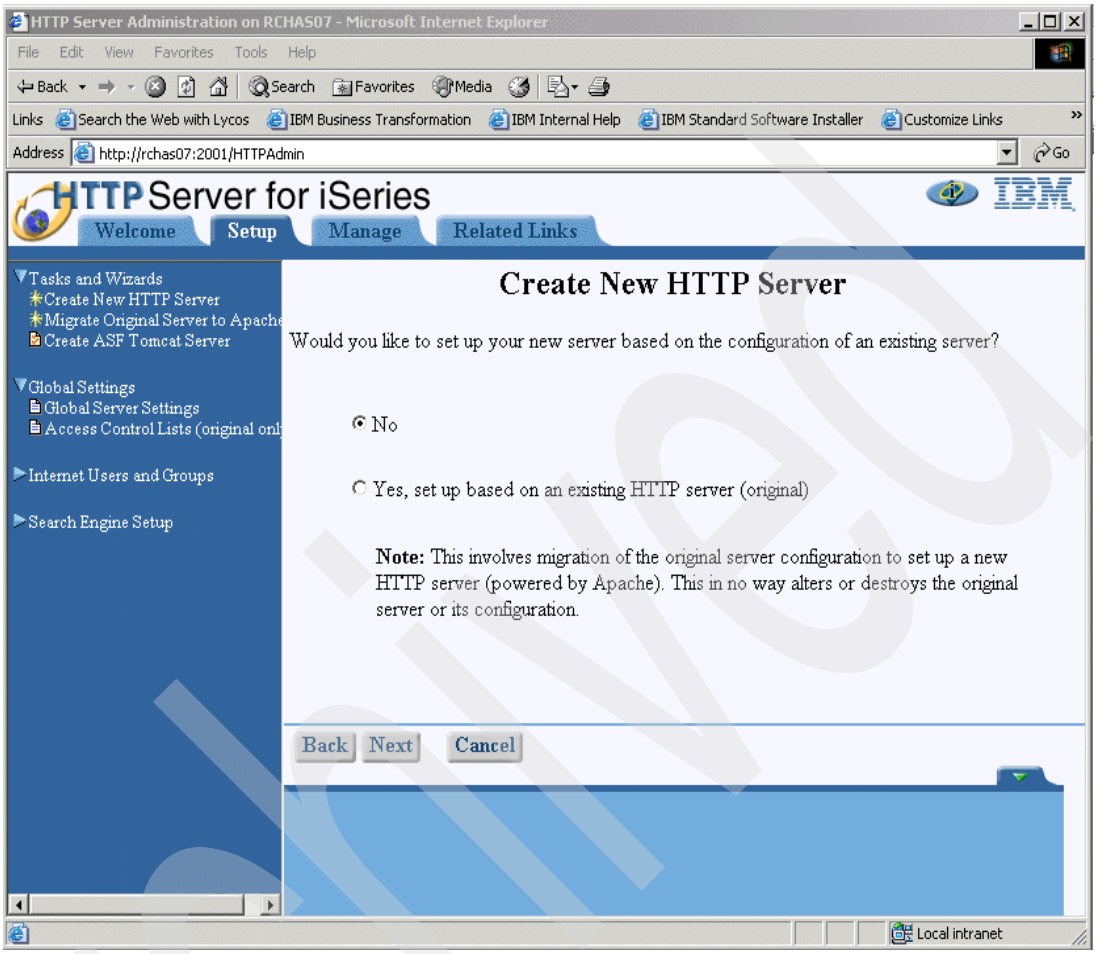


Figure 6-41 OS/400 V5R2 new configuration

8. In the next panel (see Figure 6-42), specify the name for the server root directory.
9. Click **Next**.

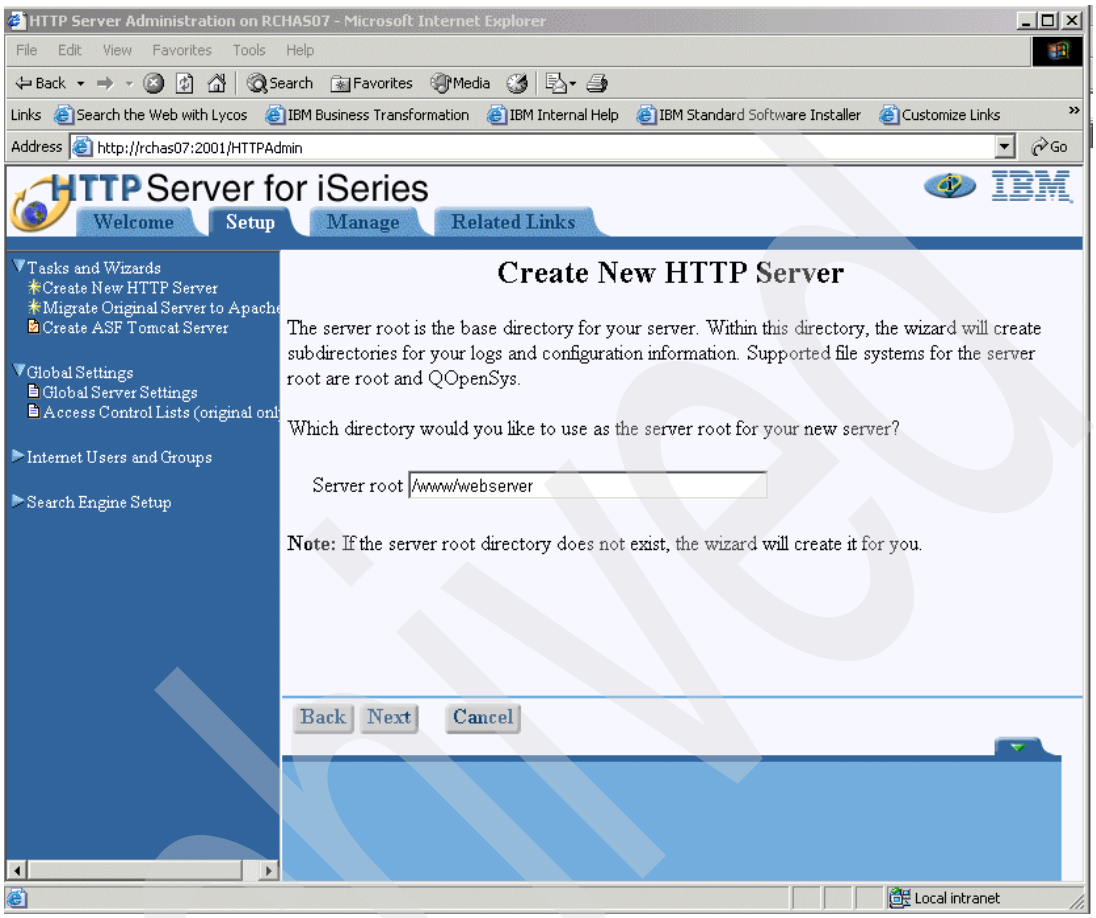


Figure 6-42 OS/400 V5R2 Define Server root

10. In the next panel (see Figure 6-43), enter the name for the document root.
11. Click **Next**.

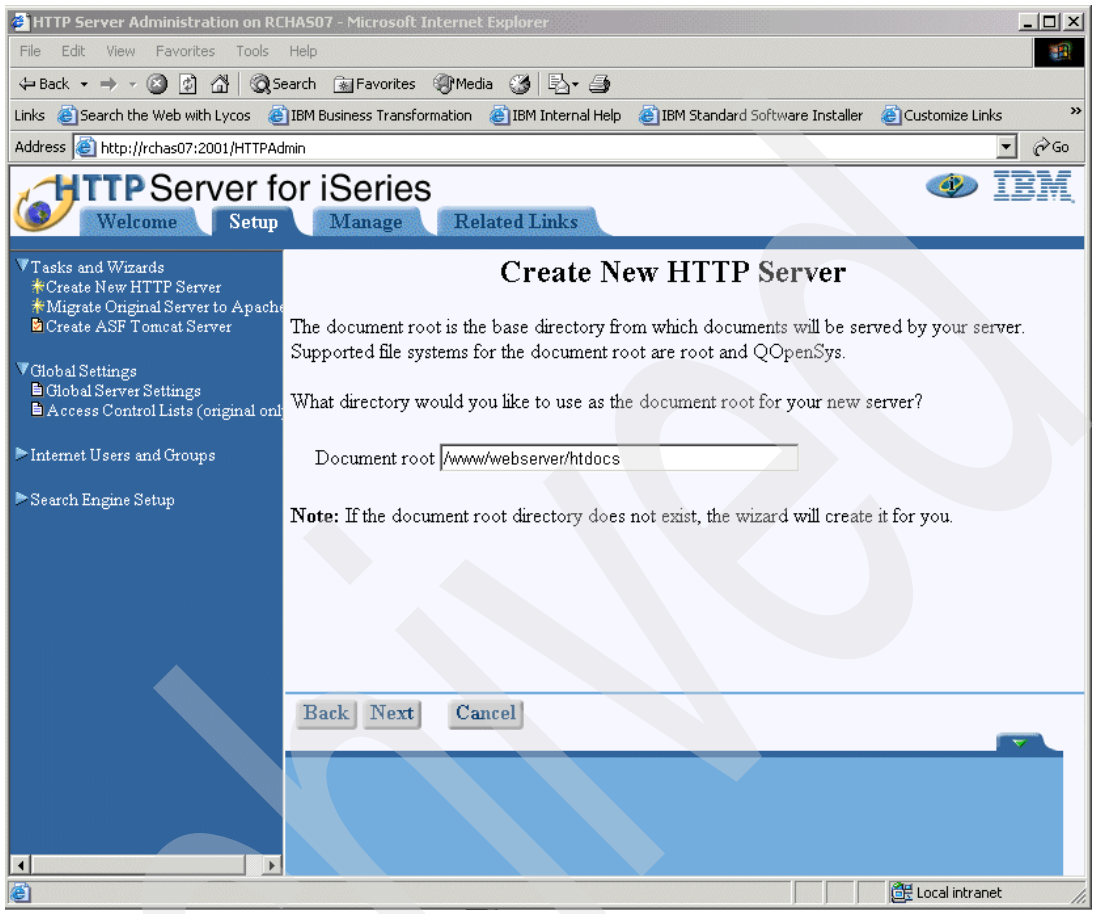


Figure 6-43 OS/400 V5R2 Define Document root

12. Select the IP address and port (see Figure 6-44):

   – In the IP address field, select **All addresses**.

     You can also select a dedicated IP address to process requests.

   – The default port is 80.

     The port number you specify here has to match the external HTTP port number that you use while creating a new WAS instance. See “Creating a new WAS instance” on page 113 for more information.

13. Click **Next**.

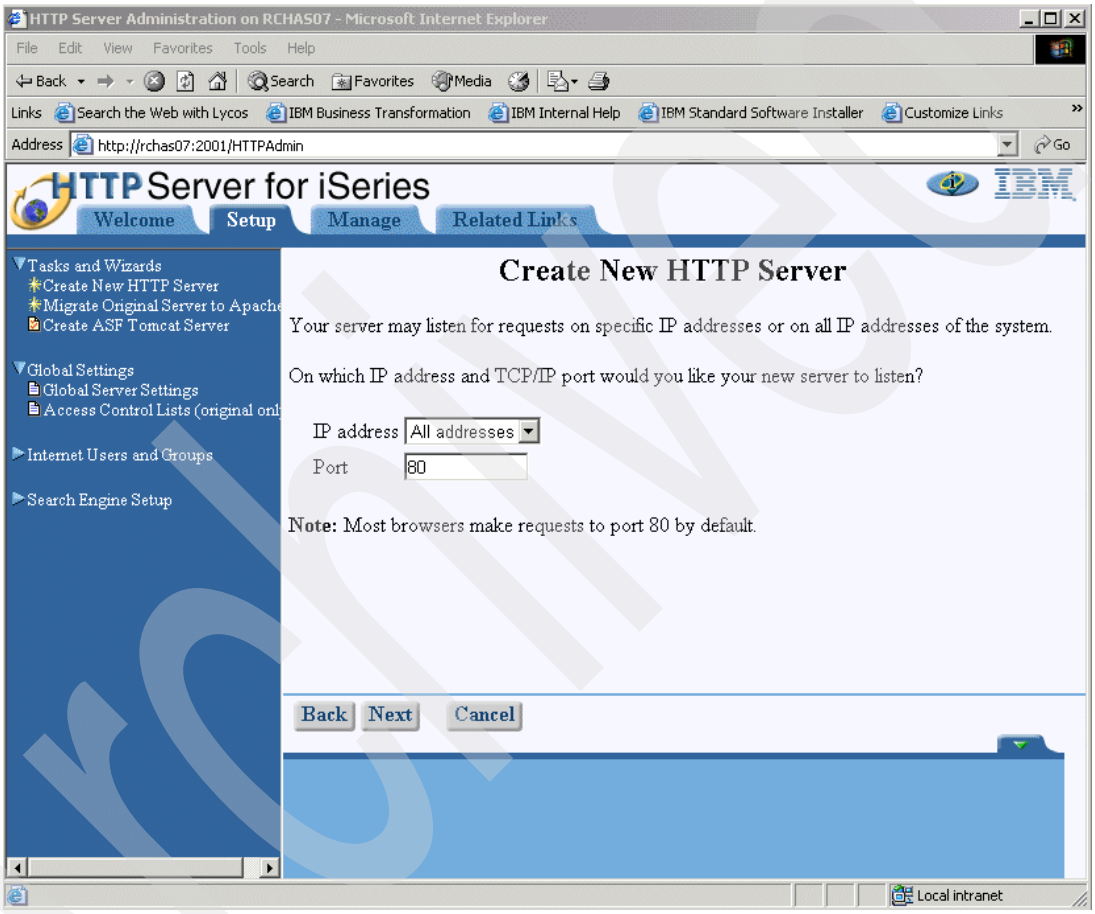


Figure 6-44 OS/400 V5R2 Select address and port

14. Select **Yes** for the server record activity on your Web site using an access log (see Figure 6-45).

    The logs are stored in the /<*server_root*>/logs directory, where <*server_root*> is the server root that you set in step 8 (see Figure 6-42 on page 149).

15. Click **Next**.

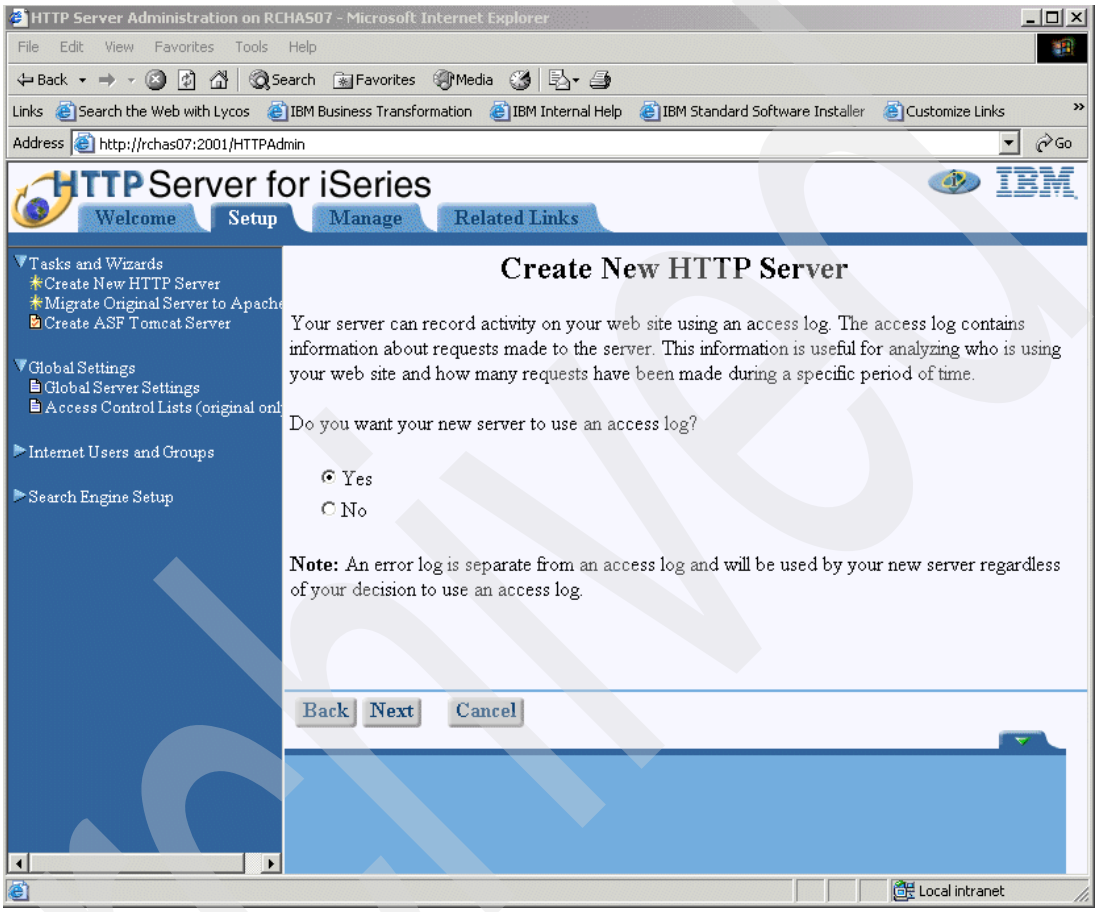


Figure 6-45 OS/400 V5R2 Combined logs

16. Review your settings which are summarized in the next screen (see Figure 6-46). To create the HTTP server instance, click **Finish**.

    To make changes, click **Back**.

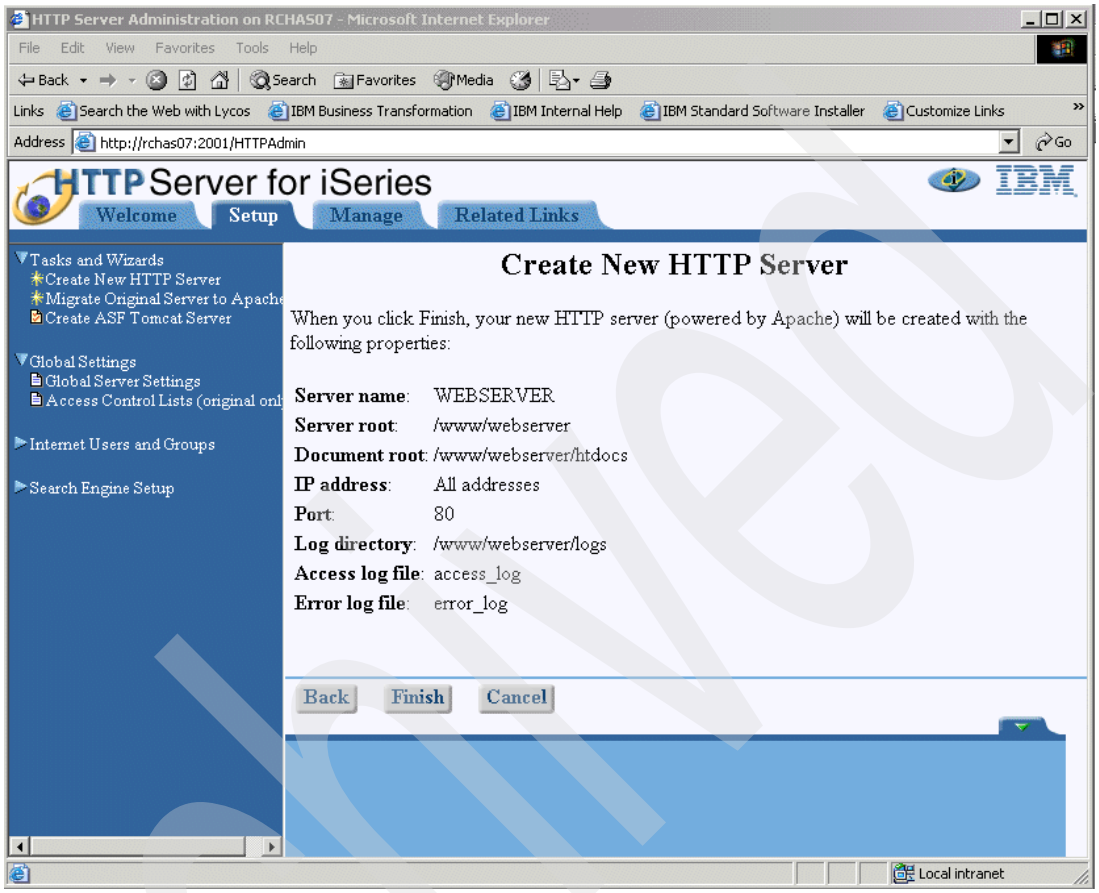


Figure 6-46 OS/400 V5R2 Configuration summary

17. The next panel acknowledges the creation of the new HTTP server.

18. After you created the HTTP server instance, configure the instance so that it can communicate with the WebSphere Application Server.

    Select **Manage newly created server** and click **OK** (see Figure 6-47).

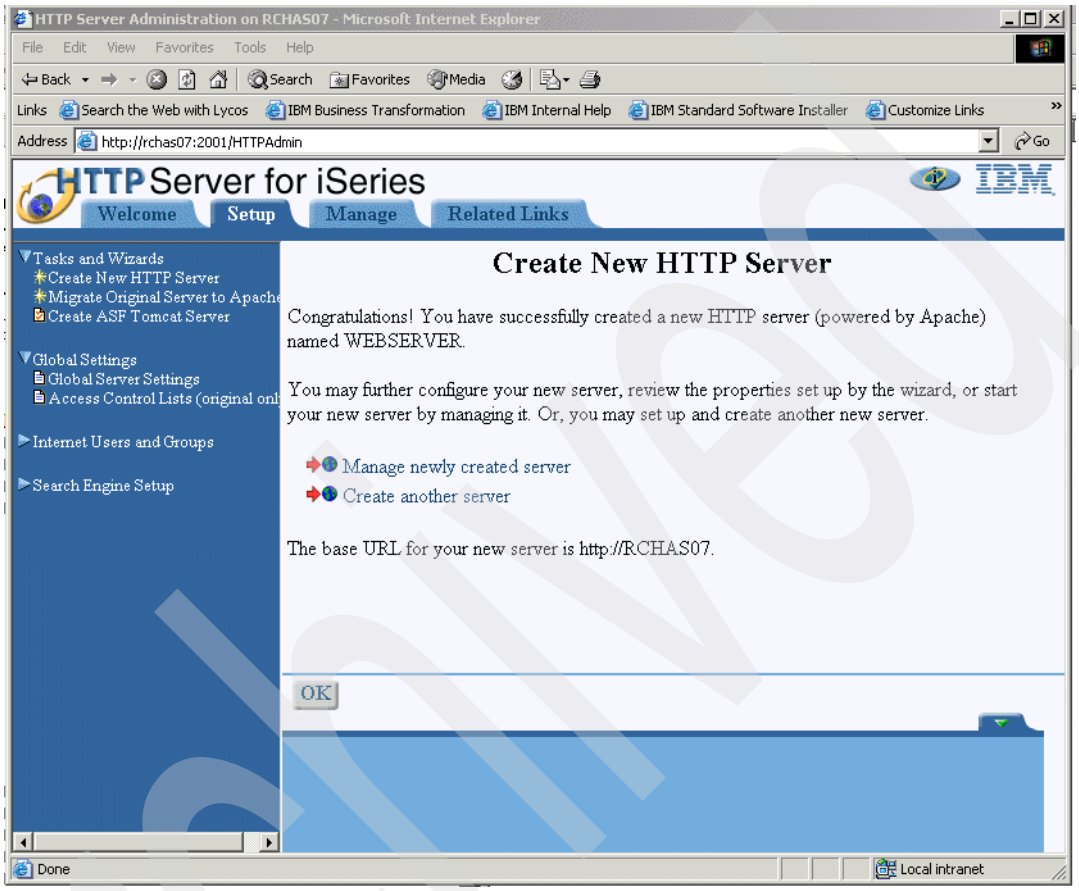


Figure 6-47 OS/400 V5R2 New HTTP server created

19. Expand **Server Properties** in the left frame and click **WebSphere Application Server** (see Figure 6-48).

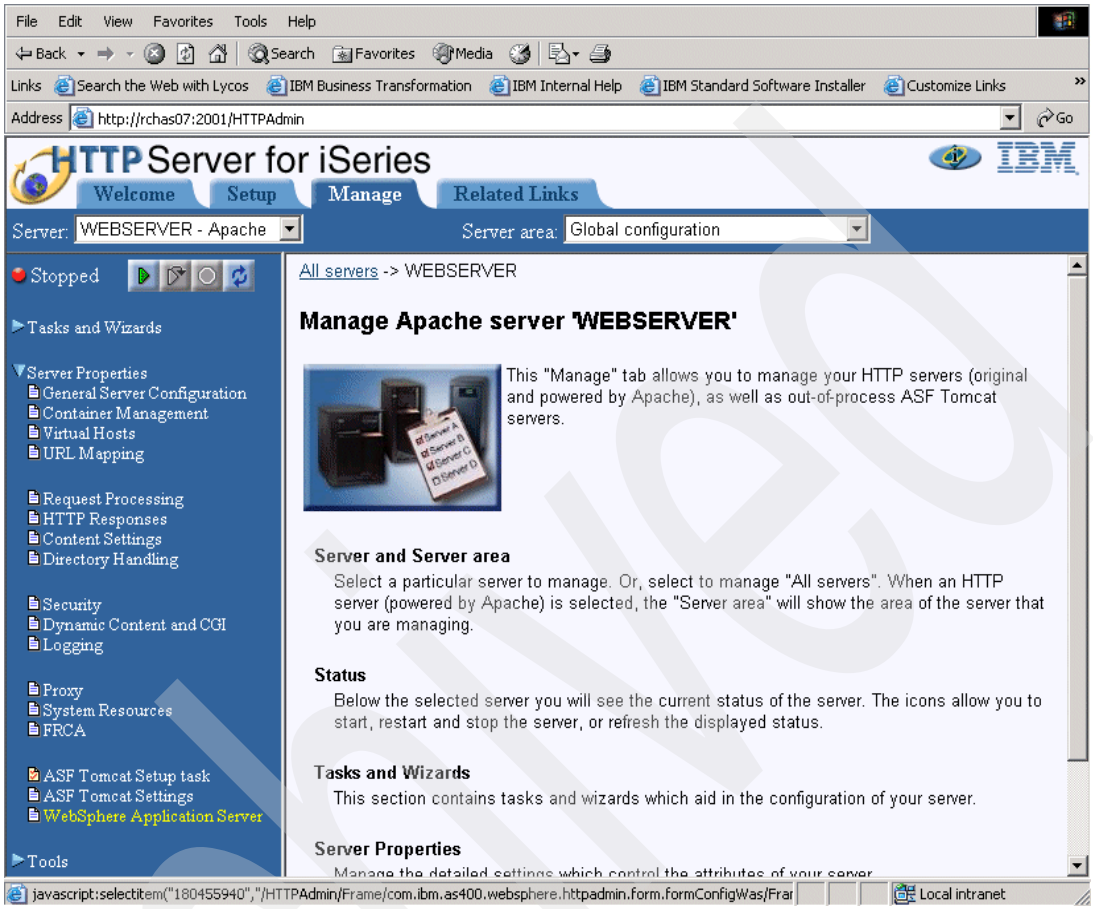


Figure 6-48 OS/400 V5R2 Manage HTTP Server

20. The selections available on the next panel (see Figure 6-49) depend on which versions of WebSphere are currently installed on the iSeries server.

21. Select **WebSphere version 5** and choose the desired WAS instance from the selection list, and click **Apply (**see Figure 6-49).

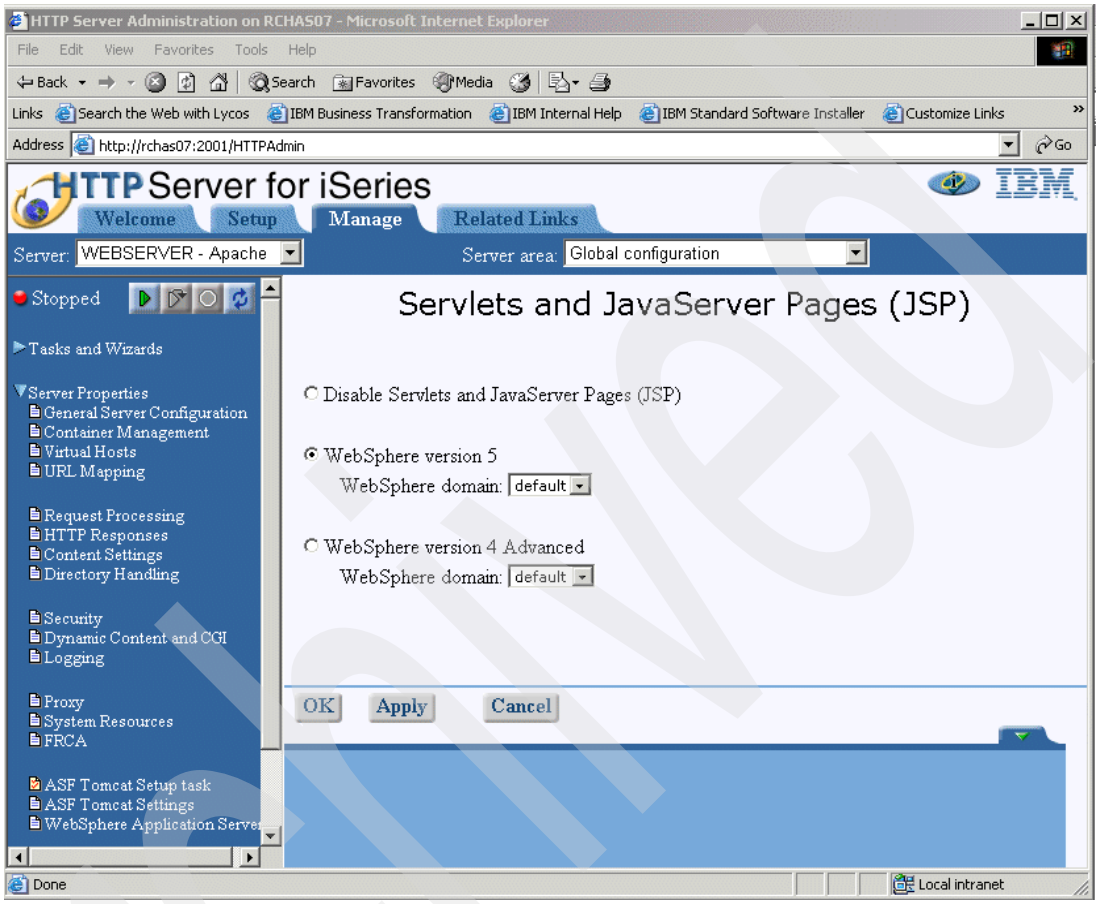


Figure 6-49 OS/400 V5R2 Select WAS version and instance

Several directives are added to the HTTP server instance configuration file. Figure 6-50 shows an example of these directives.

```
 1    <Directory /QIBM/ProdData/WebAS5/Base/WSsamples/>
 2         Allow from all
 3         Order allow,deny
 4         AllowOverride None
 5         Options None
 6    </Directory>
 7    Alias /WSsamples/ /QIBM/ProdData/WebAS5/Base/WSsamples/
 8    LoadModule ibm_app_server_http_module /QSYS.LIB/QEJBAS5.LIB/QSVTIHSAH.SRVPGM
 9    WebSpherePluginConfig
/QIBM/UserData/WebAS5/Base/default/config/cells/plugin-cfg.xml
```

Figure 6-50 HTTP server directives related to a WAS instance

22. From the next panel you can start (or restart) the HTTP server instance. Click the green start button in the left hand frame (see Figure 6-51) to start the HTTP server.

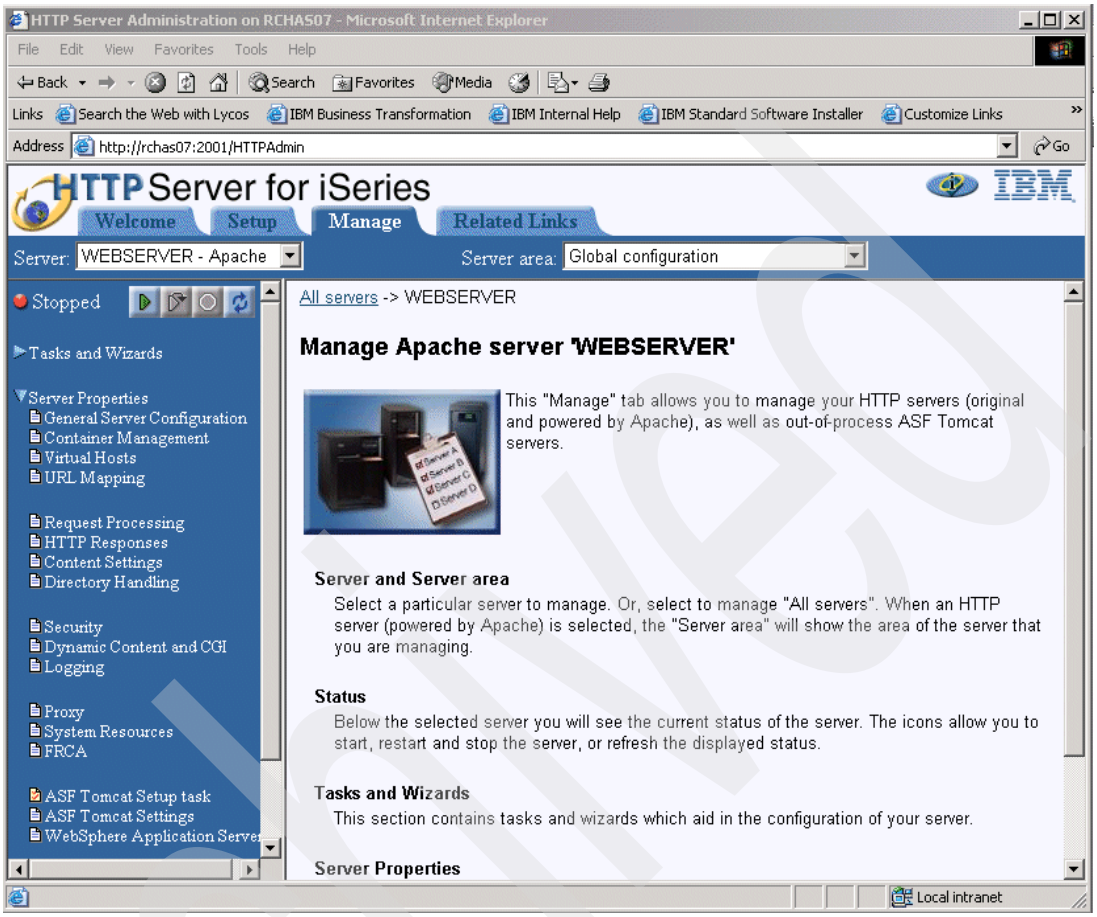


Figure 6-51 OS/400 V5R2 Manage and Start HTTP server

6.6.4 Configuring Lotus Domino Web server

Note: The ITSO team hasn't used a Domino Web server with WebSphere Application Server. The instructions in this section are copied here from the WAS InfoCenter for convenience.

To configure the Lotus Domino HTTP server for use with WebSphere Application Server, you must update the Domino server's configuration. The Lotus Domino for iSeries Version 5.0.5 (or later) HTTP server can be used as a Web server with WebSphere Application Server.

Note: Refer to the Domino 5 Administration Help for information on installing and setting up Domino servers on OS/400. The help database is shipped with Lotus Domino, and is also available in the Notes.net Documentation Library.

http://www.notes.net/notesua.nsf?OpenDatabase

To configure each instance of a Domino server on your iSeries server, follow these steps:

1. Update the Domino server document to work with WebSphere Application Server.

   To update the Domino server document to work with WebSphere Application Server, perform these steps:

   a. From a Lotus Notes client connected to the appropriate Domino server, edit the Domino document, which can be found in the Domino server's Domino Directory. The name of this file is names.nsf.

   b. Within the server document, click the **Internet Protocols** tab and then click the **HTTP** tab.

   c. Type this line in the **DSAPI filter file names** field:

      ```
      /QSYS.LIB/QEJBAS5.LIB/LIBDOMINOH.SRVPGM
      ```

   d. Save and exit the Domino Server document.

   Note: To use the Lotus Domino Web server with WebSphere Application Server, you do not have to modify the Java servlet support field in the Domino server document.

2. Update the Domino server's notes.ini file.

   To update the Domino server's notes.ini file to work with WebSphere Application Server, perform these steps:

   a. Enter the Work with Domino Servers (**WRKDOMSVR**) command on the OS/400 command line.

   b. For the appropriate Domino server instance, specify option **13** (Edit NOTES.INI) to edit the server's notes.ini file.

   c. Add this line to the end of the notes.ini file:

      ```
      WebSphereInit=/qibm/userdata/webas5/Base/default/config/plugin-cfg.xml
      ```

      Note: The line above is for the WebSphere Application Server's default instance. For a non-default instance, change the line to refer to the directory of the application server instance.

   d. Press F3 twice to save and exit the notes.ini file.

3. After you update the Domino server document and the server's notes.ini file, you must restart the Domino server's HTTP task for those changes to take effect.

   To restart the Domino server's HTTP task, perform these steps:

   a. Enter the Work with Domino Servers (**WRKDOMSVR**) command on the OS/400 command line.

   b. For your Domino server instance, specify option **8** (Work console) to select the Domino server's console.

   c. From the Domino server's console, enter the following command to restart the Domino server's HTTP task:

      ```
      tell http restart
      ```

6.7 Working with the Administrative Console

The administrative console is a browser-based graphical administrative interface for configuring and managing WebSphere resources. You can use the administrative console to display and change your WebSphere Application Server configurations, and to manage your WebSphere Application Server resources.

Important: The administrative console is implemented as a Web application. For this reason, you have to start an application server to access the administrative console. For details on starting an application server, refer to 6.5.3, “Starting a specific application server” on page 117.

The browser-based administrative console for WebSphere Application Server requires that cookies be enabled in the browser. Therefore, before you start the administrative console on your workstation, ensure that cookies are enabled.

- In Netscape Navigator, perform these steps:
  a. Click the **Edit** pull-down menu and choose **Preferences**.
  b. In the left pane, click **Advanced**.
  c. The right pane shows the settings for cookies. Ensure that cookies are enabled.
- In Microsoft Internet Explorer 4.0, 5.0, and 5.5, perform these steps:
  a. Click the **Tools** pull-down menu and choose **Internet Options**.
  b. Click the **Security** tab.
  c. Click **Custom Level**. Under **Cookies**, select **Enable**.
  d. Click **OK**.
- In Microsoft Internet Explorer 6.0, perform these steps:
  a. Click the **Tools** pull-down menu and choose **Internet Options**.
  b. Click the **Privacy** tab.
  c. Click **Advanced**. Select **Accept** to accept first-party and third-party cookies.
  d. Click **OK**.

6.7.1 Starting the administrator console

To start the administrative console on a workstation, perform these steps:

1. Make sure that your application server has been started.
2. Open this URL in your browser:

   ```
   http://your.server.name:port/admin
   ```

   Here, `your.server.name` is the TCP/IP host/domain name of the iSeries server on which your application server instance is running; see 2.9, “Starting, configuring, and verifying TCP/IP” on page 21. The port is the administration console port, as noted in the ready message in the joblog of your application server; see “Verifying that the WAS environment has started” on page 119.

   For the default WebSphere Application Server instance, the administration console port is 9090.

Here is an example of a URL for accessing the administrative console for the default server `server1` on the RCHAS02B host:

```
http://RCHAS02B:9090/admin
```

Here is another example of accessing the administrative console on the same host, RCHAS02B, but for a different application server:

```
http://RCHAS02B:10410/admin
```

3. When prompted, enter a user ID (see Figure 6-52).

Note: The user ID does not need to be an OS/400 user profile. This user ID is used only to track which users make changes to the application server configuration.

Figure 6-52 Administrative Console Login

You will see the WebSphere Application Administrative Console Version 5.0; see Figure 6-53.

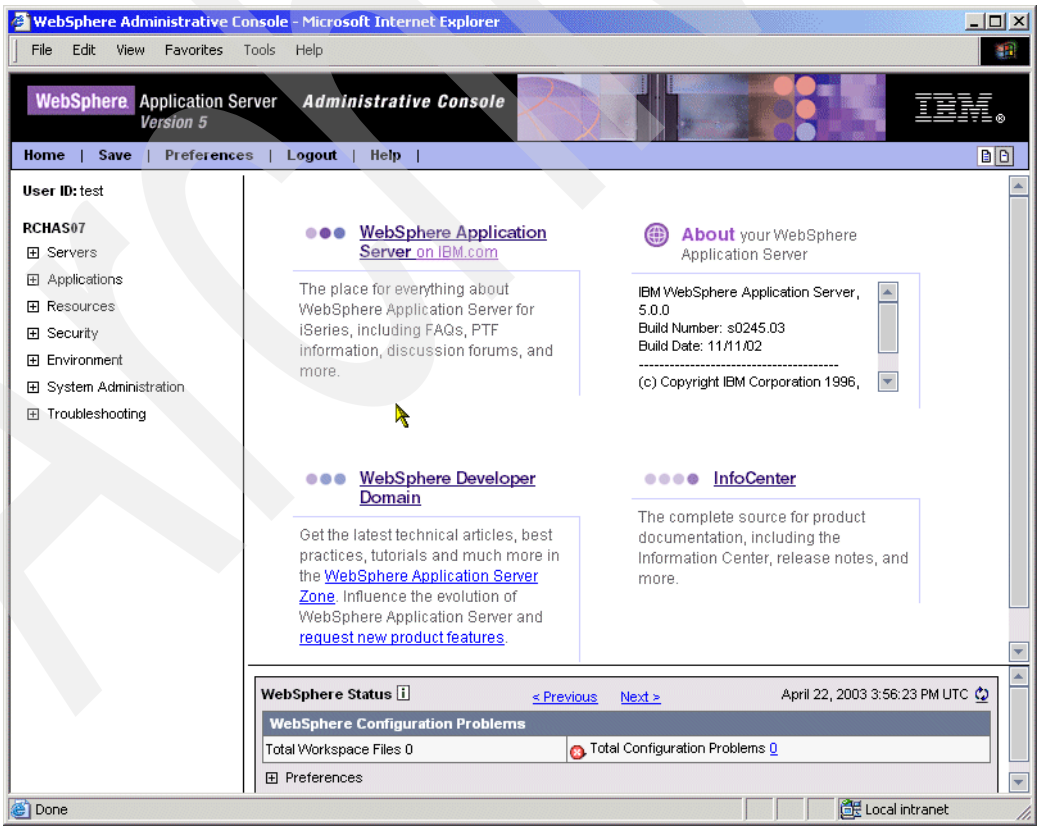


Figure 6-53 Administrative Console start

6.7.2 Configuring a virtual host

Before creating or updating a virtual host, you need to understand the role of a virtual host in the WAS environment. For this reason, we provide an overview of the virtual host concept.

A virtual host: overview

A virtual host works like a filter. It is used by the WebSphere plug-in to determine whether an HTTP request should be handled by WAS or not. A virtual host consists of one or more Domain Name Server (DNS) aliases. They are called *host aliases*. Each host alias consists of a host name and a port number. If "*" is specified as a host name, it means that all host names and IP addresses will be valid for this host alias.

A client request for a servlet, JavaServer Pages (JSP) file, or related resource contains a DNS alias and a Uniform Resource Indicator (URI) that is unique to that resource — for example, URL request `http://myhost:8080/snoop`. In this example, `myhost:8080` is the DNS alias (Host Alias), and `snoop` is the URI. When no port number is specified in the URL request, the default port 80 is used.

Note: The host aliases don't have to be the same as the host name and port number of the WebSphere Application Server(s). They are the host name(s) and port number(s) that the WebSphere plug-in is expecting to receive in the client's request.

When a client request for a servlet, JSP file, or related resource is received, the DNS alias is compared to the list of all known host aliases to locate the correct virtual host. Similarly, the URI is compared to the list of all known URIs to locate the correct resource (servlet or JSP). If a matching virtual host or URI is not found, an error is returned to the browser.

Mapping an HTTP request to host aliases and URIs is case sensitive, and the match must be alphabetically exact. Also, different port numbers are treated as different aliases.

For example, the request `http://www.myhost.com/myservlet` does *not* map to the following:

```
http://myhost/myservlet
http://www.myhost.com/MyServlet
http://www.myhost.com:9876/myservlet
```

A virtual host also maintains a list of Multipurpose Internet Mail Extensions (MIME) types that the application server will process. MIME is an Internet standard for multimedia and e-mail, including graphics, audio, and fax.

Each Web application deployed in WAS is associated with one and only one virtual host. However, one virtual host can be associated with many Web applications.

You can configure more than one virtual host. With multiple virtual hosts you can separate all your applications, deployed to a specific application server, into groups. As an example, you can group all your intranet applications under one virtual host and all Internet applications under another virtual host.

Default virtual hosts and port number

Two virtual hosts are created for each application server by the `crtwasinst` script: `default_host` and `admin_host`.

`default_host` is used for the user applications that are deployed to the application server. `admin_host` is used for the WebSphere Administrative Console.

Table 6-3 shows the default port numbers that are used by the default instance.

Note: In order to specify your own port numbers, use port related parameters for the `crtwasinst` script. Refer to “Creating a new WAS instance” on page 113 for more information.

Table 6-3 Default virtual host ports and host names

Virtual host	host name	port	HTTP server
default_host	*	80	external HTTP server
	*	9080	internal HTTP server
admin_host	*	9090	Administrative Console
	*	9043	Administrative Console via SSL

Updating Host alias table

We provide the instructions for updating the default_host virtual host, any other virtual host is updated in the same way. Follow these steps to update the host aliases table:

1. Start the administrative console for your instance; see 6.7, “Working with the Administrative Console” on page 159.
2. Expand **Environment** in the left frame of the administrative console.
3. Click **Virtual Hosts** in the left frame (see Figure 6-54).

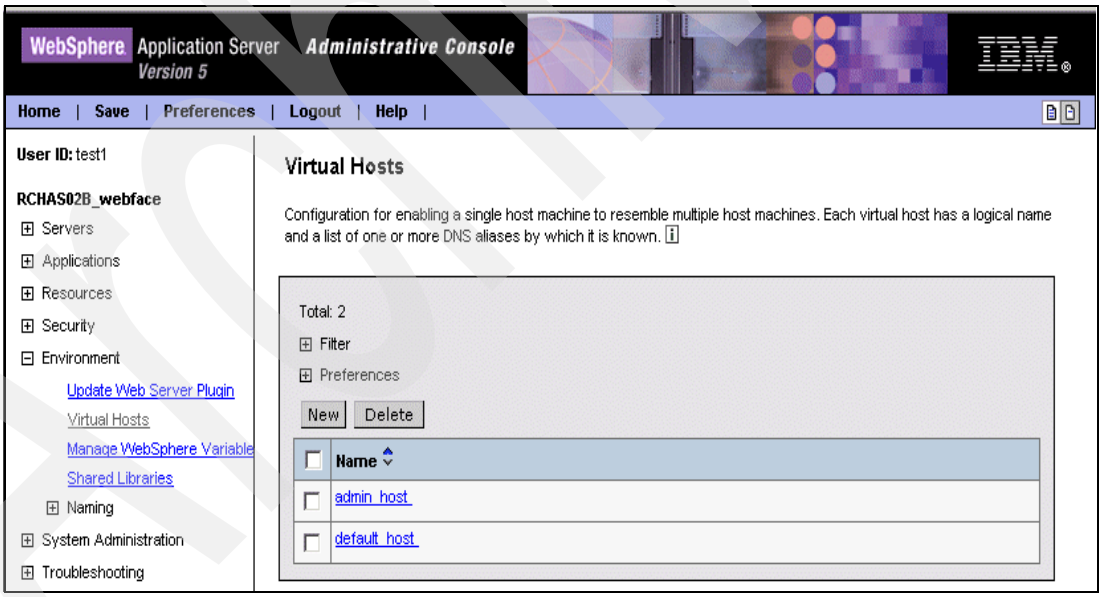


Figure 6-54 Update virtual host

4. Click **default_host** in the right frame. You will get the next panel (see Figure 6-55).
5. Click **Host Aliases** in the Additional Properties section.

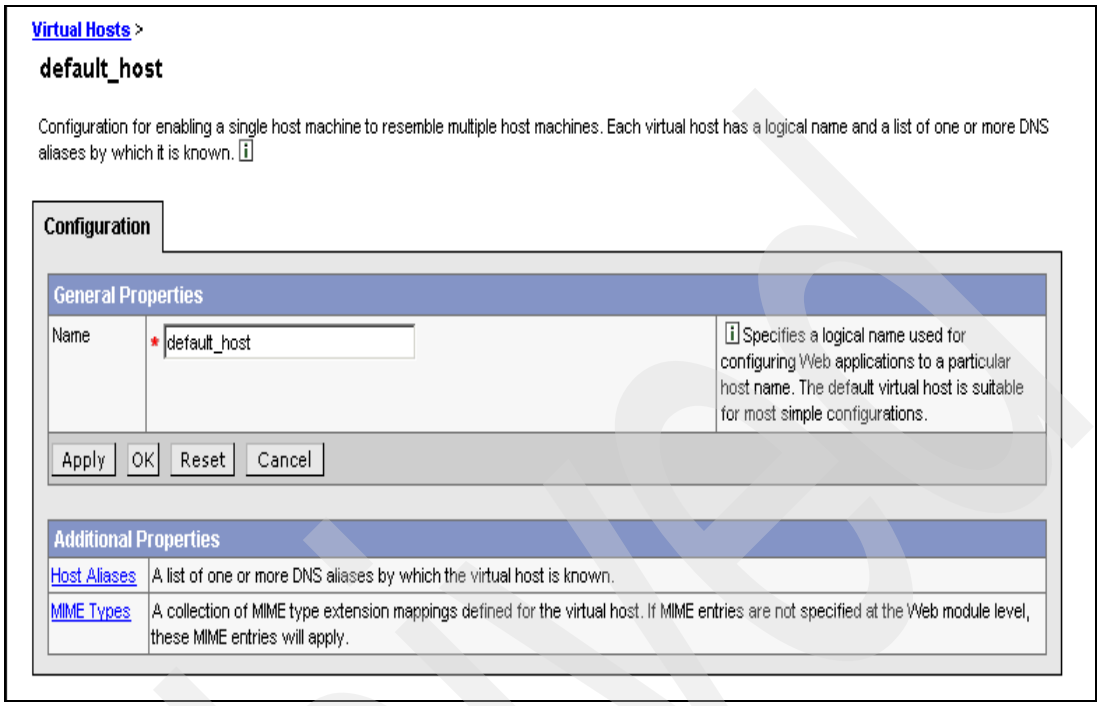


Figure 6-55 Virtual host - change alias start

6. In the Host Aliases list, click the **Host Name** link (the star in our example case) in the row that has port 80 defined (see Figure 6-56). Port 80 is allocated for the external HTTP server, and port 10409 is the application server's internal HTTP port.

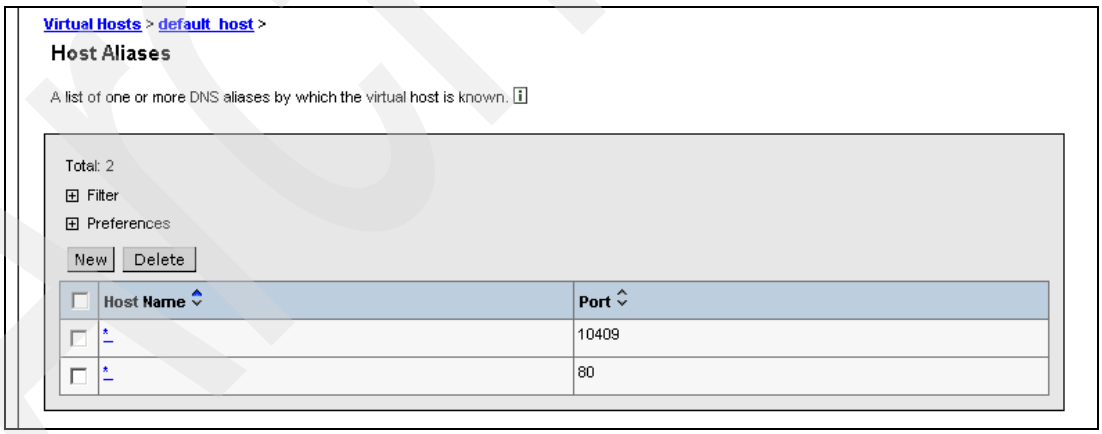


Figure 6-56 Virtual host definition for default host- instance webface

7. Change the port number from 80 to the desired value (in our example it is **10416**) and click **OK** (see Figure 6-57).

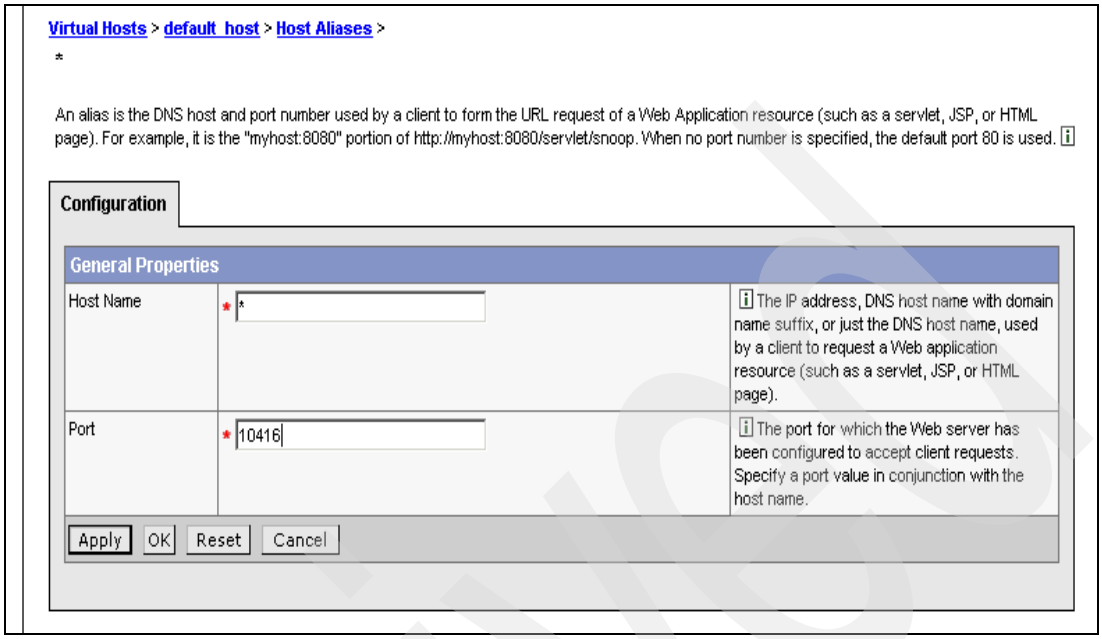


Figure 6-57 Set the external HTTP port

8. Click **Save** to save your configuration (see Figure 6-58).

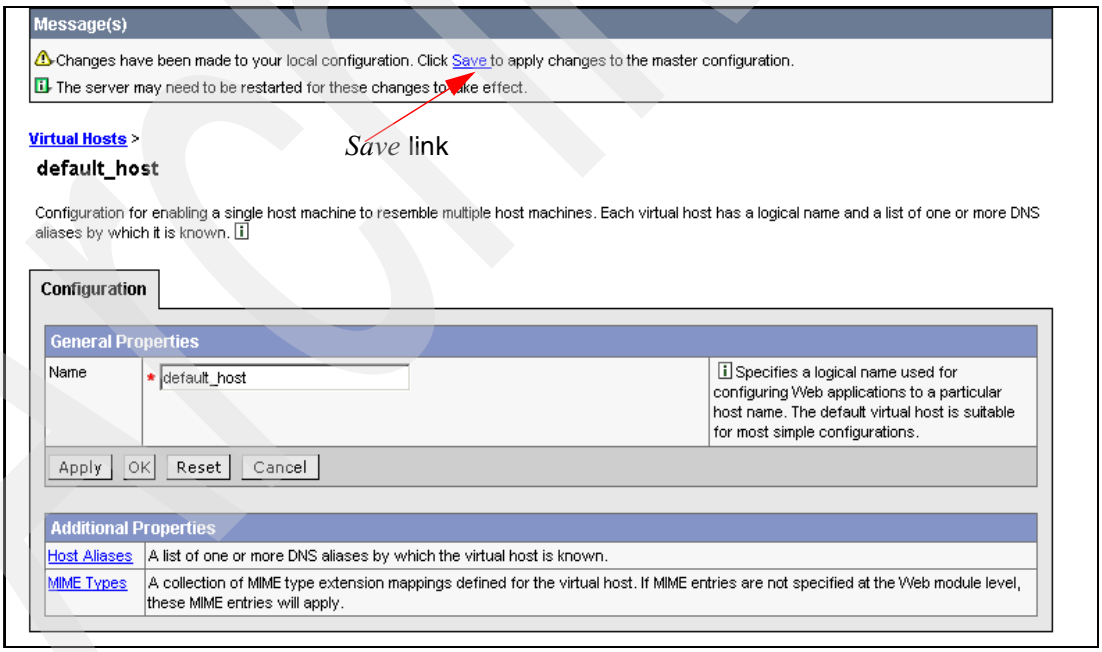


Figure 6-58 Save the changes to master configuration

9. Click **Save** to save your workspace changes to the master configuration (see Figure 6-59).

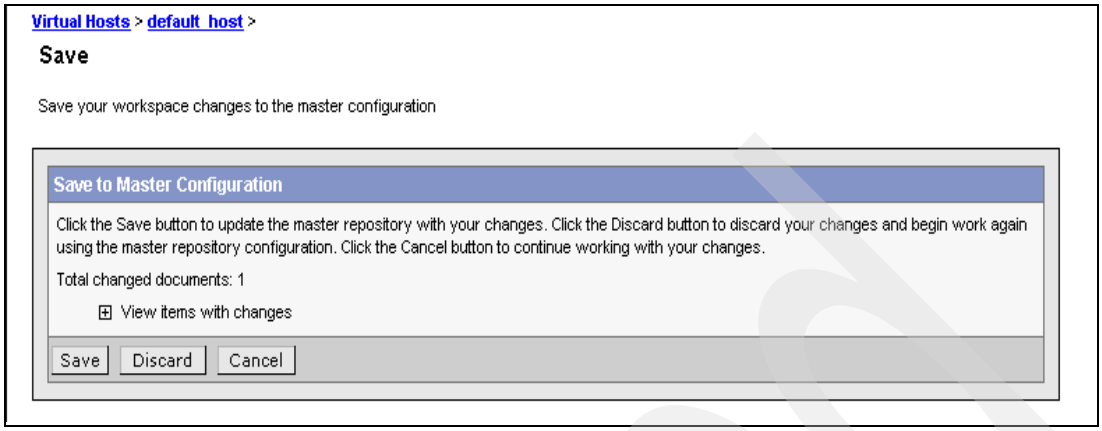


Figure 6-59 Save your workspace changes to the master configuration

10. After this change, regenerate the WebSphere plug-in configuration file (perform the steps that are described in 6.7.4, “Updating Web server plug-in configuration” on page 171).
11. Restart your WAS instance to make this changes effective; see 6.8, “Restarting the WebSphere Application Server” on page 176.

Important: Every time you change a virtual host, the application server and the corresponding HTTP server must be restarted.

Creating a new virtual host

If you need to create a new virtual host, follow these instructions:

1. Make sure that your application server is up and running (see 6.5.4, “Verifying that the WAS environment has started” on page 119).
2. Start the administrative console for your application server.
3. Expand **Environment** and click **Virtual Hosts**.

4. Click the **New** button in the **Virtual Hosts** panel (see Figure 6-60).

Figure 6-60 Creating a new virtual host

5. In the **New** panel, enter the name of the new virtual host and click **OK**.
6. You should see the name of the new virtual host in the list (see Figure 6-61).

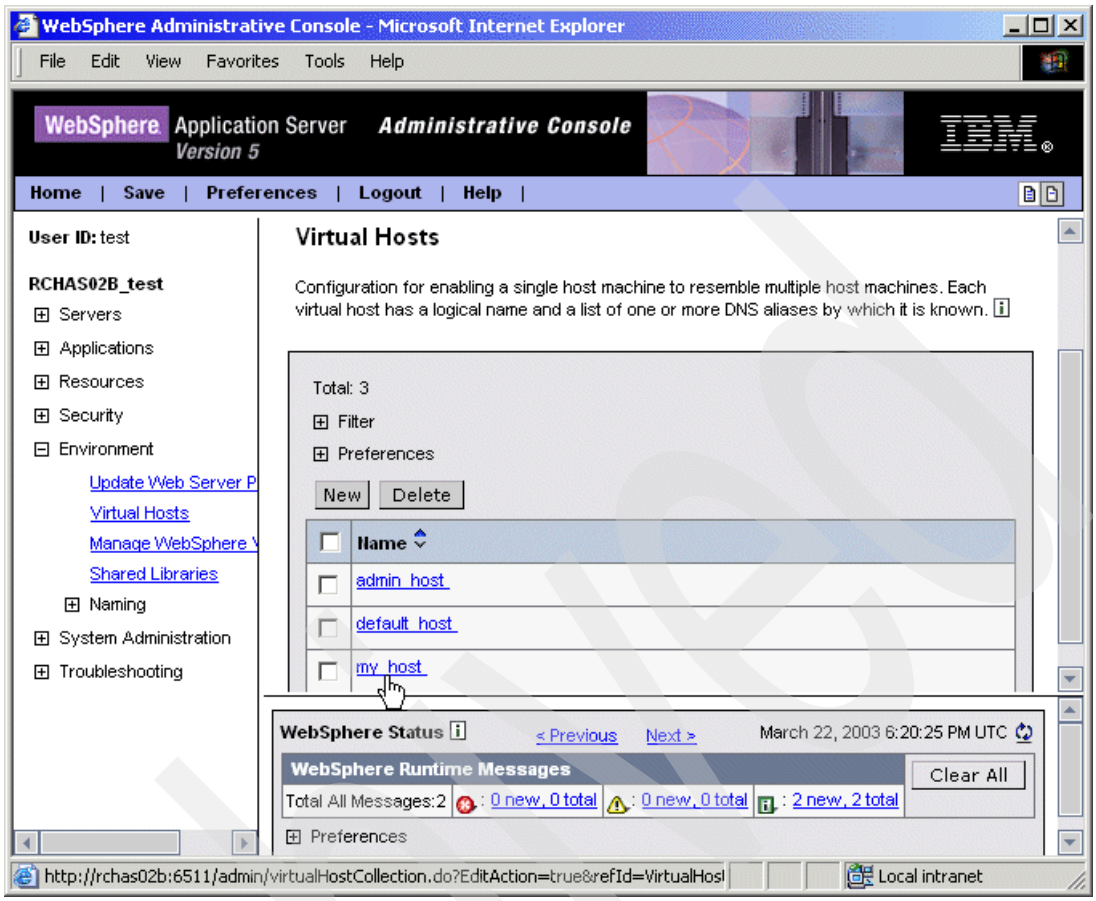


Figure 6-61 The new virtual host

7. Click the link for your new virtual host.
8. Click **Host Aliases**.
9. In the **Host Aliases** panel, click the **New** button.
10. In the next panel, enter the values for the host name and port number.
11. Click **OK**.
12. Repeat steps 8 through 11 to add another host alias.
13. When you're done, click the **Save** link at the top of the panel.
14. Click the **Save** button in the confirmation panel.
15. Regenerate the Web server plugin file (see 6.7.4, "Updating Web server plug-in configuration" on page 171).
16. Restart the application server (see 6.8, "Restarting the WebSphere Application Server" on page 176).

6.7.3 Changing Web container HTTP transport settings

A Web container is a component that provides the runtime support for the Web components. The 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.

The Web server plug-ins, provided by the WebSphere Application Server, help the supported Web servers pass servlet requests to the Web containers.

One of the parameters for a Web container is an HTTP transport.

HTTP transport

AN HTTP transport is the request queue between a WebSphere Application Server plug-in for Web servers and a Web container in which the servlet and JSPs reside. When a user at a Web browser requests an application, the request is passed to the Web server, then along the transport to the Web container.

Transports define the characteristics of the connections between a Web server and an application server, across which requests for applications are routed. Specifically, they define the connection between the Web server plug-in and the Web container of the application server.

Administering transports is closely related to administering WebSphere Application Server plug-ins for Web servers. Indeed, without a plug-in configuration, a transport configuration is of little use.

internal HTTP transport

For applications in a test or development environment (in other words, a non-production environment), you can use the internal HTTP transport system to serve servlets without a Web server plug-in. Simply use the internal HTTP transport port (the default port is 9080).

For a production environment, do not use the internal transport, as it lacks the performance as well as some of the advanced features that are available when using a Web server plug-in.

Instructions for changing the HTTP transport settings

Follow these steps to change the Web container HTTP transport settings:

1. Start the administrative console for your instance; see 6.7, “Working with the Administrative Console” on page 159.
2. Expand **Servers** in the navigation tree of the administrative console.
3. Click the application server link that you want to change (see Figure 6-62).

Figure 6-62 Change HTTP Transport - start

4. Click **Web Container** (see Figure 6-63).

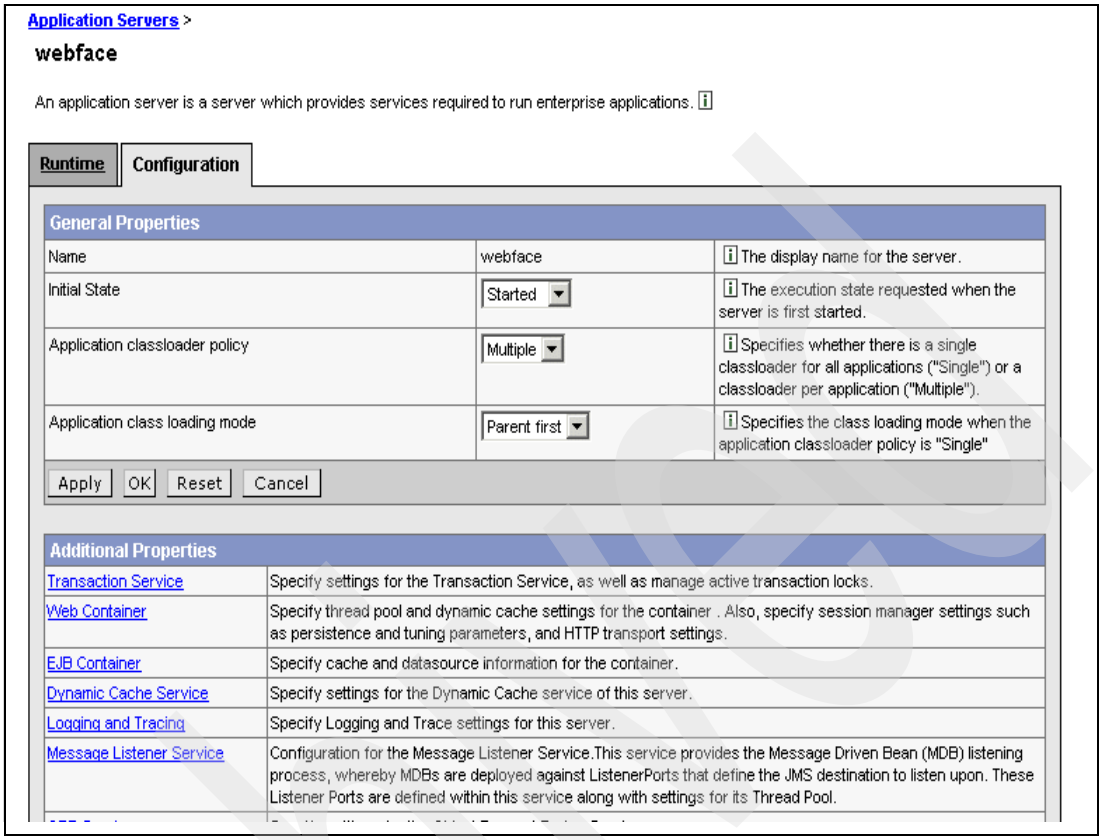


Figure 6-63 Change Web container settings

5. On the next panel, click **HTTP transports** (see Figure 6-64).

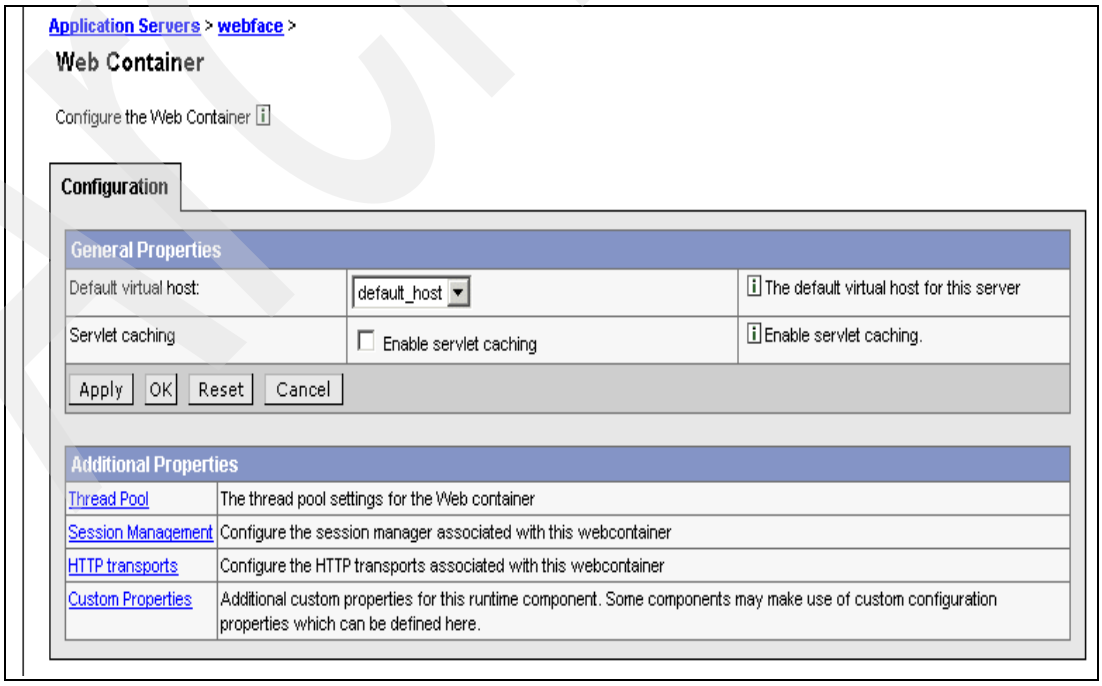


Figure 6-64 HTTP transports

6. Figure 6-65 shows the HTTP transport setting for our example: the internal HTTP port 10409, the admin port 10410, and the admin SSL port 10411 (they were assigned when we created the instance with the `-portblock` parameter; see Figure 6-2 on page 115). We want to change the internal HTTP port number from 10409 to 10415.
7. Click the **Host** link on the line for the port you want to change (in our case the * in the line with port number 10409).

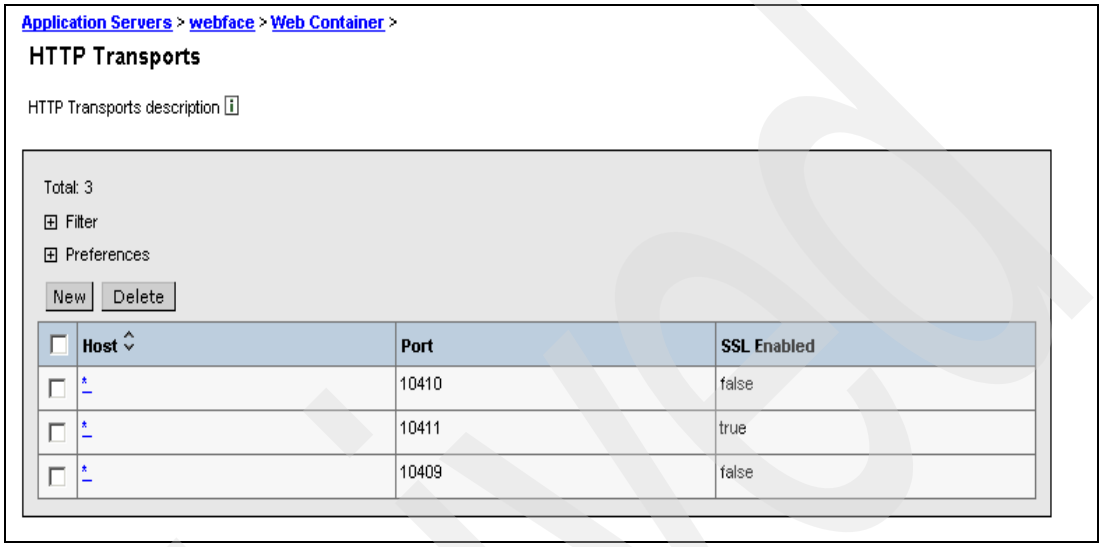


Figure 6-65 Internal and admin ports for instance webface

8. Type the new port number (see Figure 6-66) and click **OK**.

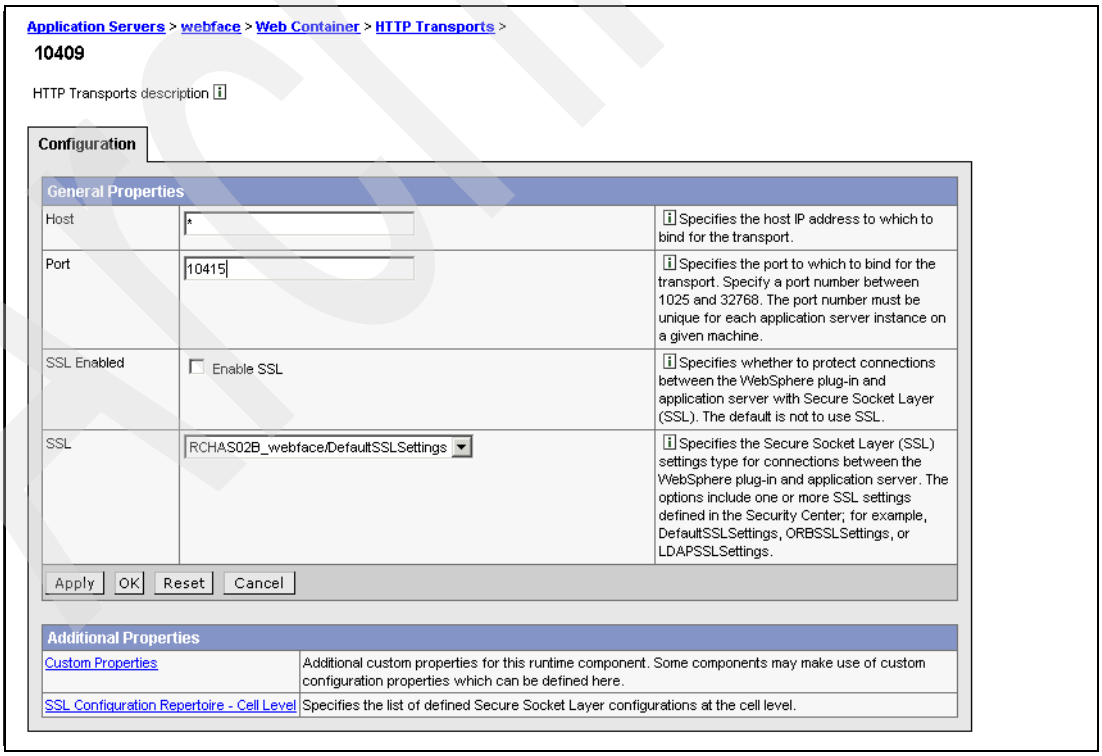


Figure 6-66 Change internal HTTP port

9. Click **Save** to save your local configuration; see Figure 6-67.

Figure 6-67 HTTP transport list

10. Click **Save** to save your workspace changes to the master configuration.

Important: Make sure you update the virtual host with the new HTTP port number.

11. After this change, regenerate the Web server plug-in (perform the steps that are described in 6.7.4, “Updating Web server plug-in configuration” on page 171).
12. Also, you have to restart your WAS instance to make these changes effective (see 6.8, “Restarting the WebSphere Application Server” on page 176).

6.7.4 Updating Web server plug-in configuration

A WebSphere application server works with an HTTP server to handle requests for Web applications. The HTTP server and application server communicate using a WebSphere plug-in for the Web server. The WebSphere plug-in component uses a special file, the WebSphere plug-in configuration file, to get the configuration details of WebSphere Application Server. The WebSphere plug-in configuration file (plugin-cfg.xml) controls what content is passed from the Web server to an application server.

In the following cases, the plug-in configuration file must be regenerated:

- Adding/removing a new URL (Web resource or Web module) to the environment
- Securing or unsecuring a URI
- Configuring a new virtual host alias
- Changing the port number for a host alias
- Adding a new application server to the environment

The WebSphere plug-in configuration file (plugin-cfg.xml) is placed in the configuration directory of the WebSphere Application Server instance. It is stored in the directory `/QIBM/UserData/WebAS5/Base/instance_name/default/config/cells`, in the IFS of your iSeries server (`instance_name is` the name of your instance).

If you need to regenerate the WebSphere plug-in configuration file, perform these steps (you can also use the `GenPluginCfg` script):

1. Start the administrative console from a browser; see 4.8, “Start the Administrative Console on Windows” on page 127
2. Expand **Environment**.
3. Click **Update Web Server Plug-in**. You will see a panel as shown in Figure 6-68.

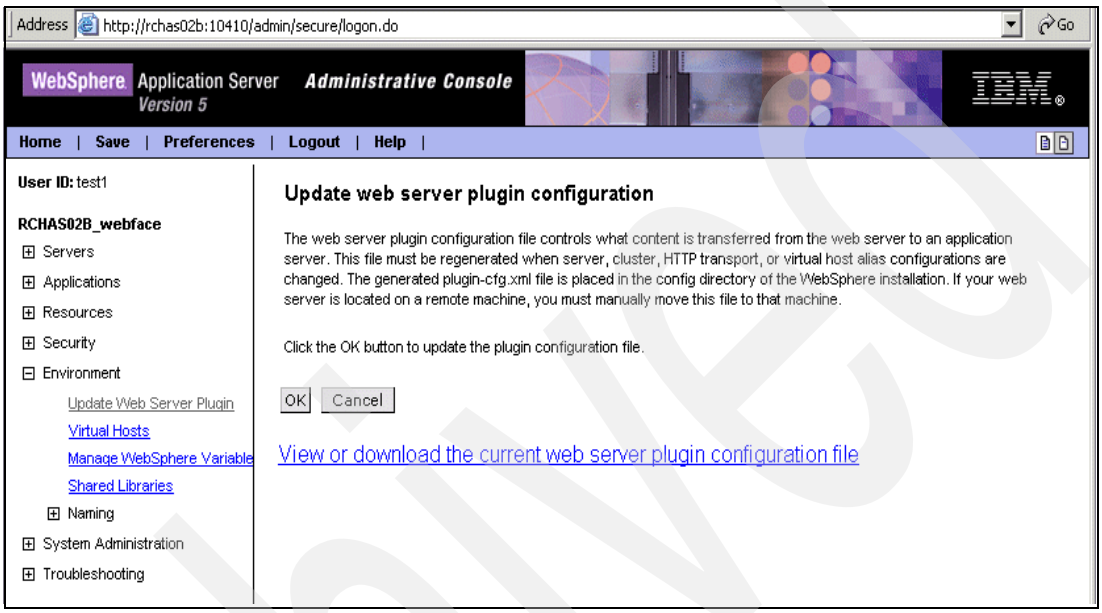


Figure 6-68 Update Web server plug-in

4. Click **OK**.

   You will seen a confirmation message like the one shown in Figure 6-69.

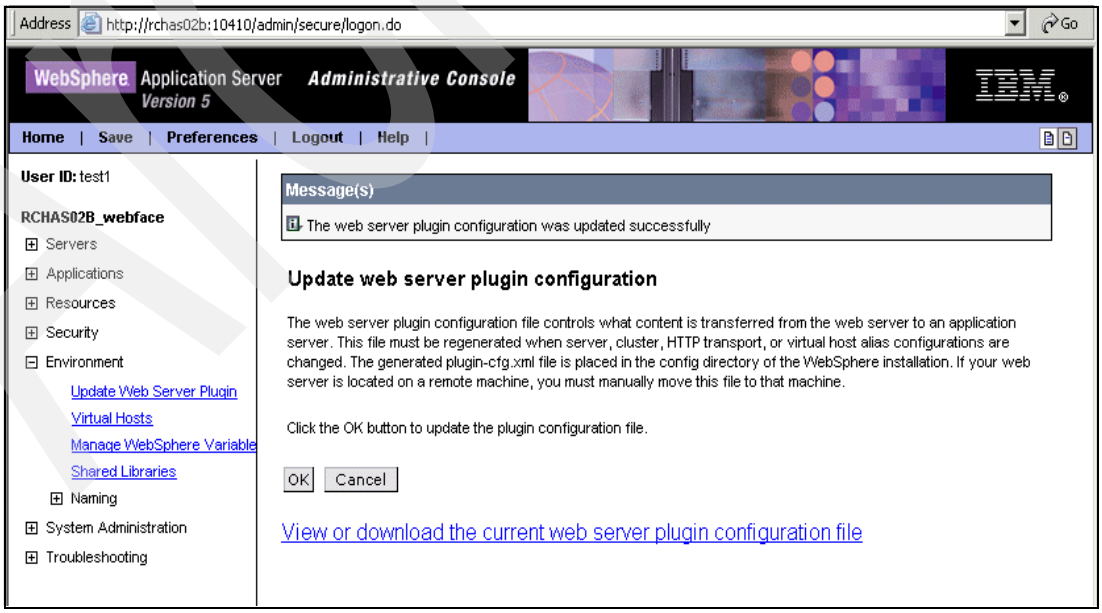


Figure 6-69 Update Web server plug-in confirmation

5. You can use the link in the right frame to view the current WebSphere plug-in configuration file (plugin-cfg.xml). Figure 6-70 shows an example of this file.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Config>
    <Log LogLevel="Error"
Name="/QIBM/UserData/WebAS5/Base/webface/logs/http_plugin.log"/>
    <VirtualHostGroup Name="default_host">
        <VirtualHost Name="*:10420"/>
        <VirtualHost Name="*:10416"/>
    </VirtualHostGroup>
    <ServerCluster Name="webface_RCHAS02B_webface_Cluster">
        <Server Name="webface">
            <Transport Hostname="RCHAS02B" Port="10416" Protocol="http"/>
        </Server>
        <PrimaryServers>
            <Server Name="webface"/>
        </PrimaryServers>
    </ServerCluster>
    <UriGroup Name="default_host_webface_RCHAS02B_webface_Cluster_URIs">
        <Uri AffinityCookie="JSESSIONID" Name="/snoop/*"/>
        <Uri AffinityCookie="JSESSIONID" Name="/hello"/>
        <Uri AffinityCookie="JSESSIONID" Name="/hitcount"/>
        <Uri AffinityCookie="JSESSIONID" Name="*.jsp"/>
        <Uri AffinityCookie="JSESSIONID" Name="*.jsv"/>
        <Uri AffinityCookie="JSESSIONID" Name="*.jsw"/>
        <Uri AffinityCookie="JSESSIONID" Name="/j_security_check"/>
        <Uri AffinityCookie="JSESSIONID" Name="/servlet/*"/>
        <Uri AffinityCookie="JSESSIONID" Name="/ivt/*"/>
        <Uri AffinityCookie="JSESSIONID" Name="/WebFacing1/*"/>
    </UriGroup>
    <Route ServerCluster="webface_RCHAS02B_webface_Cluster"
        UriGroup="default_host_webface_RCHAS02B_webface_Cluster_URIs"
VirtualHostGroup="default_host"/>
</Config>
```

Figure 6-70 plugin-cfg.xml for instance WEBFACE after installing an application

Notice that the following information is included in the plug-in file:

- The location of the log file for the WebSphere plug-in component
- Host aliases
- The internal HTTP transport information
- URIs for the deployed applications (the line in bold in Figure 6-70 represents the custom application)

6.7.5 iSeries unique administrative console features

When compared to versions of WebSphere Application Server that run on other platforms, the WebSphere Application Server for iSeries treats the following configuration objects and properties of objects (all of which are found in the administrative console) in a unique way.

- Application servers:

  a. *Name*: This name is also used for the iSeries job name, and is listed on the Configuration tab of the Application Servers page. iSeries job names are restricted to 10 characters in length, and must consist of alphanumeric and a few special characters. If you specify an application server name that is not consistent with these iSeries job name restrictions, the job name will compress leading and trailing blanks and replace unsupported characters with an underscore.

  b. *Runtime*: The Runtime tab displays runtime attributes of an application server. This tab is only visible if the selected application server is running.

  c. *Process ID*: This property indicates the process ID of a server. For a node on an iSeries server, this is the qualified job name of the application server.

  d. *Java Virtual Machine*: The Java Virtual Machine (JVM) page contains advanced settings for an application server's JVM. To view this page, click **Process Definition** on the application server's page, then click **Java Virtual Machine**.

     It is recommended to specify the initial heap size. Start with the default value of 96MB, and increase it if needed.

     Maximum heap size on OS/400 should always be left unspecified (or 0).

  e. *Process Execution*: The Process Execution page contains advanced process execution settings for an application server. To view this page, click **Process Definition** on the application server's page, then click **Process Execution**.

     i. *Run As User*: This property specifies the iSeries user profile under which the application server job runs. This property is displayed on the Configuration tab of the Process Execution page. If left blank, the default is to run under the QEJBSVR user profile. If you specify an iSeries user profile name other than the default user profile name (QEJBSVR) for the application server User ID property, the following attributes for the application server job are set to match the settings for the specified user profile name:

        - Coded character set identifier
        - Country or region identifier
        - Current library
        - Character identifier control
        - Initial library list
        - Job accounting code
        - Language identifier
        - Locale
        - Output queue name
        - Output queue priority
        - Print text
        - Sort sequence
        - Status message handling

     Note: If you specify an iSeries user profile other than QEJBSVR to run the application server, we recommend that you specify QEJBSVR as group profile for that user profile.

     ii. *Run As Group*: This property is ignored on iSeries.

     iii. *Run In Process Group*: This property is ignored on iSeries.

  f. *Performance monitoring service*: The Performance monitoring service page specifies settings for performance monitoring, including enablement and monitoring levels. To view this page, click **Performance monitoring service** on the application server's page.

- JDBC Provider:

  These JDBC drivers are supported for WebSphere Application Server for iSeries:

  - DB2 UDB for iSeries (Native - V5R2 and later)
  - DB2 UDB for iSeries (Native XA - V5R2 and late)
  - DB2 UDB for iSeries (Native - V5R1 and earlier)
  - DB2 UDB for iSeries (Native XA - V5R1 and earlier
  - DB2 UDB for iSeries (Toolbox)
  - DB2 UDB for iSeries (Toolbox XA)

  The Classpath and Implementation Classname are filled in automatically based on the JDBC provider that you select, and should not be changed.

- DataSource:

  To view the properties of a DataSource, select the JDBC provider that includes that DataSource. On the JDBC provider's page, click **Data Sources**, then click the name of the DataSource.

  a. *Data source Helper Classname*: This property is filled in automatically based on the JDBC provider, and should not be changed.

  b. *Custom Properties*: The custom properties that are required and supported vary for each JDBC driver. For a list of custom properties for the IBM Toolbox for Java JDBC driver see iSeries Information Center:

     - For V5R1:

       http://publib.boulder.ibm.com/iseries/v5r1/ic2924/index.htm?info/rzahh/javadoc/JDBCProperties.html

     - For V5R2:

       http://publib.boulder.ibm.com/iseries/v5r2/ic2924/index.htm?info/rzahh/javadoc/JDBCProperties.html

- NodeAgent Server:

  Note: Node agent servers are applicable only in the Network Deployment environment.

  - *Runtime*:

    The Runtime tab displays runtime attributes of a node agent. This tab is only available if the selected node agent server is running:

    - *Process ID*: This property indicates the process ID of a node agent job. For a node on an iSeries server this is the qualified job name of the node agent job.

- Deployment Manager:

  Note: Deployment Managers are applicable only in the Network Deployment environment.

  - *Runtime*: The Runtime tab displays runtime attributes of a deployment manager. This tab is only available if the selected deployment manager is running:

    - *Process ID*: This property indicates the process ID of a deployment manager. For a node on an iSeries server this is the qualified job name of the deployment manager job.

6.8 Restarting the WebSphere Application Server

There is no script to restart the WebSphere Application Server, so perform the `stopServer` and `startServer` scripts to restart your application server (see 6.5.6, “Stopping an application server” on page 124 and 6.5.3, “Starting a specific application server” on page 117).

6.9 Starting IBM HTTP Server for iSeries

You can start and stop an HTTP server instance in two different ways:

- Use an OS/400 command from an OS/400 commend line.
  a. To start the HTTP server instance from an OS/400 command line, follow these steps:
     - Enter the following command:

       ```
       STRTCPSVR SERVER(*HTTP) HTTPSVR(<myinst>)
       ```

       Here, <myinst> is the name of your HTTP server instance.

       Example: To start the HTTP server named `Webface`, we use the command:

       ```
       STRTCPSVR SERVER(*HTTP) HTTPSVR(webface)
       ```

  b. To stop the HTTP server instance from an OS/400 command line, follow these steps:
     - Enter the following command:

       ```
       ENDTCPSVR SERVER(*HTTP) HTTPSVR(<myinst>)
       ```

       Here, <myinst> is the name of your HTTP server instance.

       Example: To stop the HTTP server named Webface, we use the command:

       ```
       ENDTCPSVR SERVER(*HTTP) HTTPSVR(webface)
       ```

- Use the Configuration and Administration forms via the browser interface. Steps are provided for OS/400 V5R2:
  a. Access the IBM HTTP Server Configuration and Administration page:
     - Perform steps **1 on page 144** through **3 on page 134**.
  b. Click the **Manage** tab.
  c. Select **All servers** in the **Server** pull-down menu in the upper-left corner of the window.
  d. In the **Manage All Servers** frame, select the HTTP server you want to manage.

e. Click **Start** to start the HTTP server (see Figure 6-71).

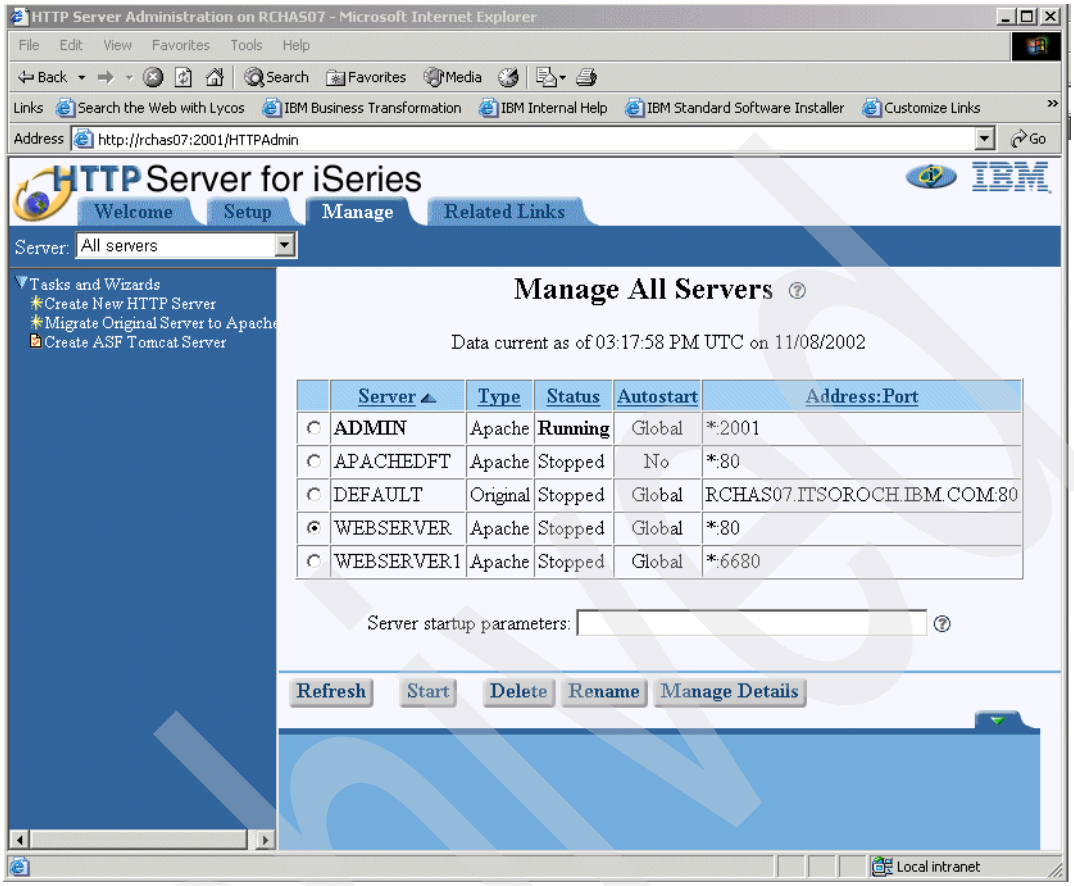


Figure 6-71 OS/400 V5R2 Manage all servers

f. Click **Refresh** to see the current status of the HTTP server.

g. If you need to stop the HTTP server instance, select the instance and click the **Stop** button (see Figure 6-72).

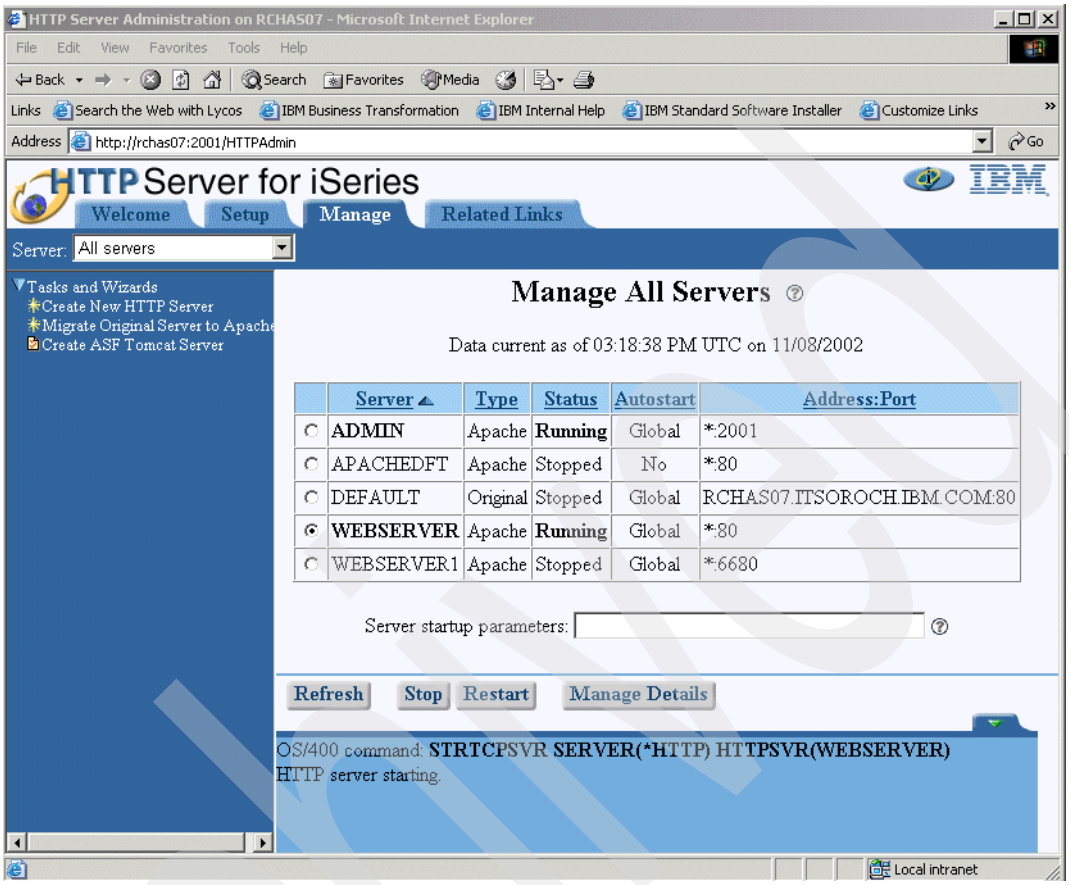


Figure 6-72 OS/400 V5R2 Manage all servers - start HTTP server instance

6.9.1 Verifying that the IBM HTTP Server for iSeries is started

All instances of the IBM HTTP Server for iSeries are running in the QHTTPSVR subsystem, and each HTTP server instance starts multiple jobs. To verify that your HTTP server instance has been started, you have two choices:

- Use the browser-based GUI interface (see Figure 6-72).
- Use an OS/400 command from an iSeries command line.

Use the Work with Active Jobs (WRKACTJOB) command from an iSeries command line, specifying the QHTTPSVR subsystem on the (SBS) parameter, and press Enter.

```
WRKACTJOB SBS(QHTTPSVR)
```

You will see a panel that is similar to Figure 6-73. Search for your HTTP server (you may have to scroll down in the panel).

If you can find any jobs with the name of your HTTP server, then the server is running (see Figure 6-73).

```
                          Work with Active Jobs                       AS07
                                                              12/08/02  17:29:48
CPU %:    3.0     Elapsed time:   00:00:23     Active jobs:   289

Type options, press Enter.
  2=Change   3=Hold   4=End   5=Work with   6=Release   7=Display message
  8=Work with spooled files   13=Disconnect ...

Opt  Subsystem/Job  User        Type  CPU %  Function        Status
     QHTTPSVR       QSYS        SBS      .0                 DEQW
      WEBFACE       QTMHHTTP    BCH      .0  PGM-QZHBHTTP    SIGW
      WEBFACE       QTMHHTTP    BCI      .0  PGM-QZSRLOG     SIGW
      WEBFACE       QTMHHTTP    BCI      .0  PGM-QZSRLOG     SIGW
      WEBFACE       QTMHHTTP    BCI      .0  PGM-QZSRHTTP    SIGW
      WEBFACE       QTMHHTTP    BCI      .0  PGM-QZSRHTTP    SIGW



                                                                    Bottom
Parameters or command
===>
F3=Exit   F5=Refresh       F7=Find      F10=Restart statistics
F11=Display elapsed data   F12=Cancel   F23=More options   F24=More keys
```

Figure 6-73 Active jobs for WEBFACE HTTP server

6.10 Verifying installation using the external IBM HTTP server

Testing with the external HTTP server is very similar to the testing using the internal HTTP server (see “Verifying installation using one of the sample Web applications” on page 122), except for two steps:

1. Make sure your external HTTP server is configured for WAS and started.
2. Use the external HTTP port in all URLs to access the applications (see 6.5.5, “Verifying the installation” on page 122).

6.11 Working with multiple application servers in an instance

Important: Even though we describe the process of creating an additional application server for a WAS instance, it is *recommended* that you create an additional WAS instance instead. There are two reasons for this:

- It is easier to manage two instances than two application servers in the same instance.
- In a cell environment (Network Deployment domain), adding two instances to a cell is easier that adding an instance with two application servers.

You can create additional application servers in the existing instance by using the administrative console of your primary application server. The primary application server is the application server that is created when you create the instance with the `crtwasinst` script; see “Creating a new WAS instance” on page 113.

The following steps are involved in the process of creating an additional application server for the existing instance:

1. Create a new application server using the administrative console of the primary application server.
2. Install the administrative console application in the new application server. With the administrative console of the primary application server, you have limited capabilities of managing the new application server: you can't, for example, start and stop applications deployed to the new application server.
3. Change the virtual hosts with the host name and port numbers of the new application server.
4. Access the administrative console of the new application server (the one that you've installed in the previous step).

6.11.1 Creating an additional application server to an instance

In order to create an additional application server, perform these steps:

1. Start the administrative console for your primary instance:

   In our case, it is `http://rchas02b:10302/admin`.
2. Expand **Servers** in the navigation tree.
3. Click **Application Servers**.
4. Click the **New** button (see Figure 6-74).

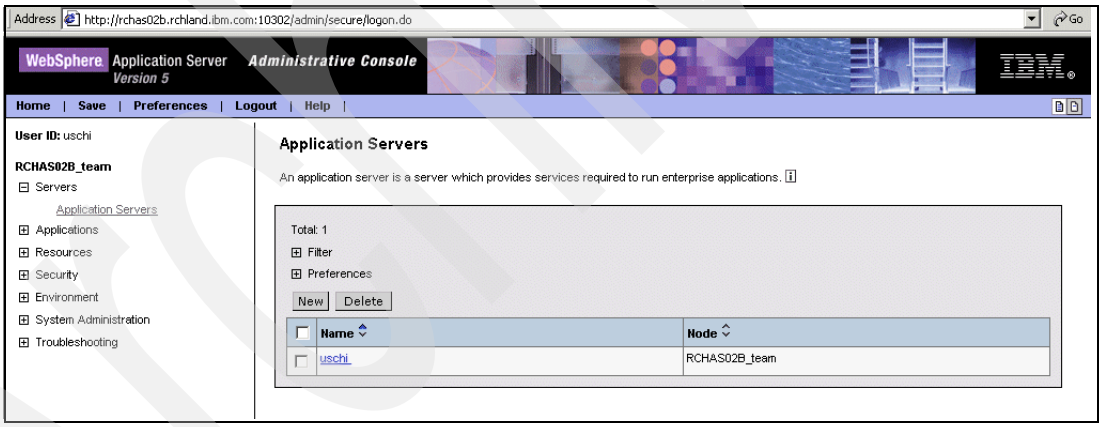


Figure 6-74 Application servers

5. Type the server name.
6. Make sure that **Generates Unique HTTP Ports** is checked (see Figure 6-75).
7. Select a template from the Select template pull-down menu. You can create the new server based on an existing application server or based on the default application server. Using the existing application server as a template will basically make a copy of the configuration of that server.
8. Click **Next**.

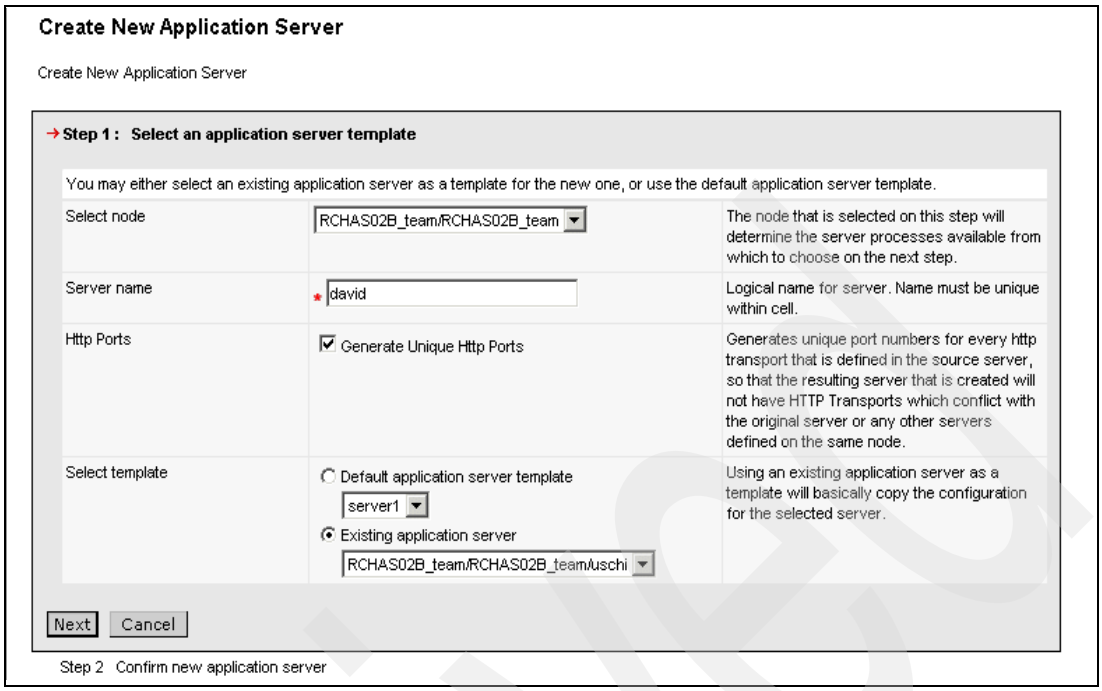


Figure 6-75 New application server

9. Review the **Confirm New Application Server** page (see Figure 6-76).

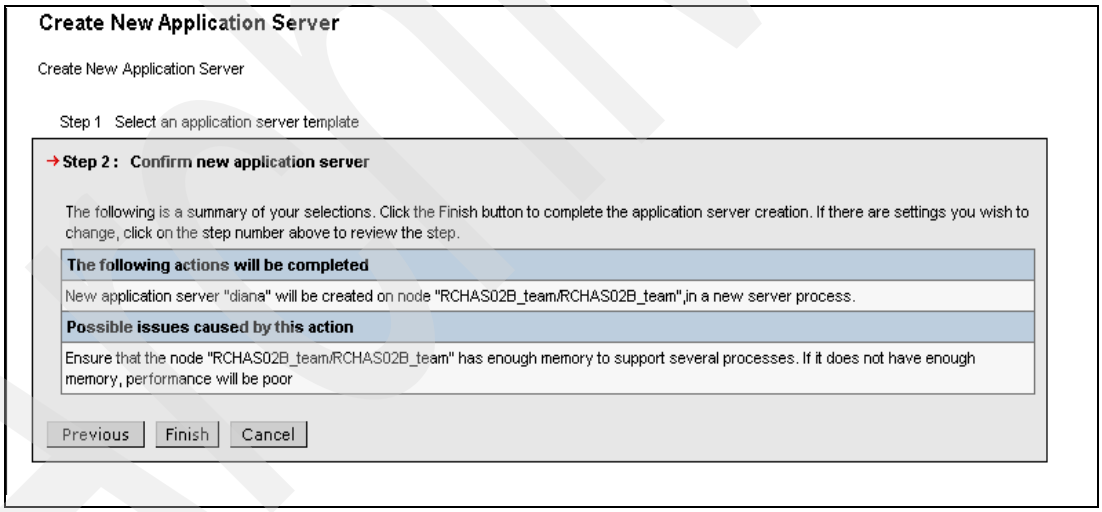


Figure 6-76 Confirm Create new application server

10. Click **Finish**.

11. Click **Save** to save your configuration (see Figure 6-77).

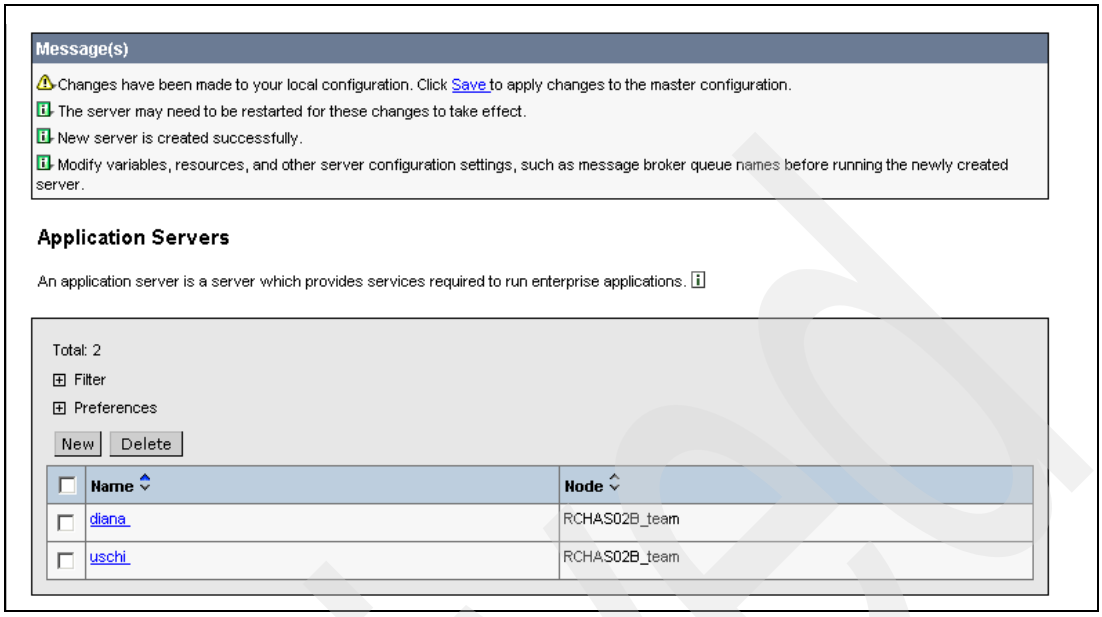


Figure 6-77 Save your configuration

12. If you expand **View items with changes**, you can see the names of XML configuration files that will be updated (see Figure 6-78).

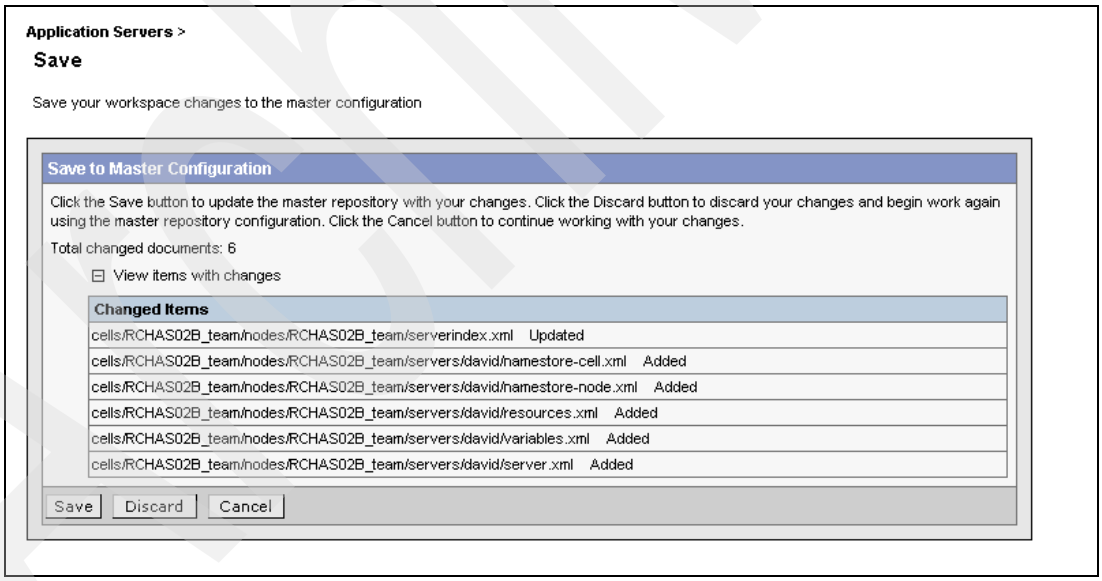


Figure 6-78 Save workspace to master configuration

13. Click **Save** to save your workspace changes to the master configuration.

14. By using the `dspwasinst` script in QShell, you can see the configuration of your instance. Figure 6-79 shows the output of this command on our system.

```
dspwasinst -instance team
ADCP0005I: Using cell RCHAS02B_team, node RCHAS02B_team and server *ALL.
Display WAS instance:
  Instance name: team
  Instance type: Base Application Server
  Cell: RCHAS02B_team
  Node: RCHAS02B_team

  Information for server: uschi
    Installed applications:
      DefaultApplication
      ivtApp
      adminconsole
    Ports in use:
        10302   Administrative console port
        10314   Administrative console SSL-enabled port
        10305   Name service port
        10310   Soap port
        10301   Internal HTTP port
        10309   Data replication service client port
        10306   Internal Java Message Service server port
        10307   Queued Java Message Service server port
        10308   Direct Java Message Service server port
        10311   SAS SSL server authentication listener port
        10313   CSIV2 server authentication listener port
        10312   CSIV2 mutual authentication listener port

 Information for server: diana
   Installed applications:
     None
     Ports in use:
         2810   Name service port
         8881   Soap port
         7874   Data replication service client port
         5558   Internal Java Message Service server port
         5559   Queued Java Message Service server port
         5560   Direct Java Message Service server port
         0   SAS SSL server authentication listener port
         0   CSIV2 server authentication listener port
         0   CSIV2 mutual authentication listener port
```

Figure 6-79 dstwasinst instance team

15. To make our new application server usable, we have to add the internal HTTP port for application server (in our example `diana` with 9081) to the virtual host configuration. Refer to 6.7.2, “Configuring a virtual host” on page 161 for the details of changing a virtual host.

16. Start the new application server by using the `startServer` script (see “Running the startServer script” on page 117).

6.11.2 Deleting the additional application servers

You can remove application servers from an instance by using the administrative console of the primary application server (you cannot remove the primary application server, which is created when you create the instance).

To remove an application server, follow these steps:

1. Stop the application server if it is running. To verify that the application server job has ended, run the Work with active Jobs command on an OS/400 command line:

   ```
   WRKACTJOB SBS(QEJBAS5)
   ```

   Verify that the application server job is not shown in the subsystem.
2. Start the administrative console for your primary instance.
3. In the navigation tree, expand **Servers** and click **Application Servers** (see Figure 6-80.
4. On the Application Servers page, select the checkbox for the application server you want to remove.

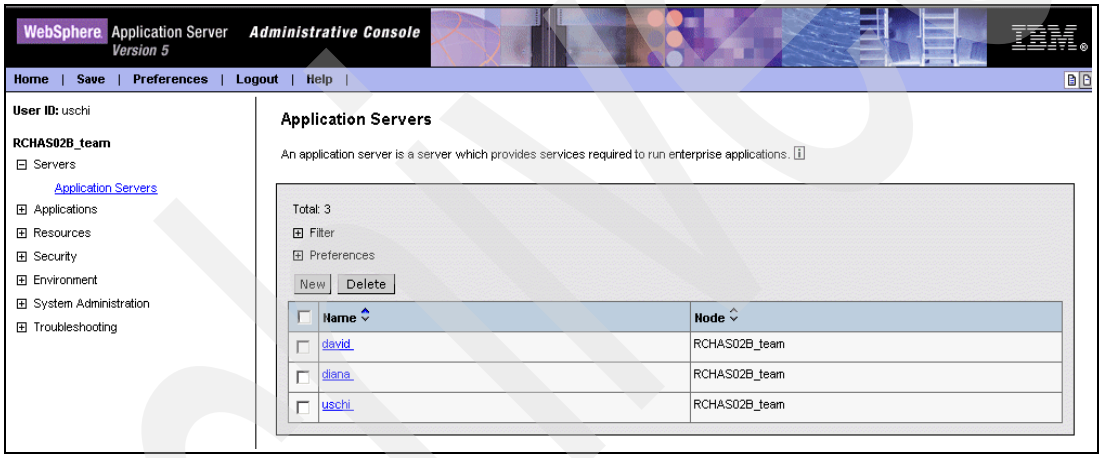


Figure 6-80 Remove additional application server

5. Click **Delete**.
6. To confirm that you want to delete the application server, click **OK**.
7. **Save** the configuration.

6.12 WebSphere Application Server samples gallery

The WebSphere Application Server samples gallery provides a set of small, generic samples that show how to perform common enterprise application tasks. The samples demonstrate the use of session and entity enterprise beans, JDBC access, connection pooling, Java Mail, message driven beans, and other Web techniques and reusable components. These samples are included in the samples gallery:

- Technology samples: This includes many smaller samples brought together in a common application.
- The Plants By WebSphere: This is an online e-store for buying plants and flowers.
- The Message Driven Beans application: This demonstrates the use of message driven beans for point-to-point and publish-subscribe messaging.

- The Petstore application from Sun Microsystems: This demonstrates how to use the capabilities of the J2EE V1.3 platform to develop flexible, scalable, cross-platform enterprise applications.

WebSphere Application Server V5.0 for iSeries provides the `installsamples` script, which does the following operations:

- It creates the database collections for the samples.
- It installs the sample applications into the specified server.
- It creates the associated resources such as JDBC drivers and DataSources.
- It updates the required properties files for the samples.

The WebSphere Application Server samples gallery is largely self-explanatory; however, we provide some helpful points to keep in mind.

By default, the samples specify *LOCAL for the database to connect to and use collections named TECHSAMP50, PLANTSDB50, PETSDB50, and CATALOG50, and several tables within these collections.

The enterprise applications in WebSphere Application Server are organized into EJB modules and Web modules. You can find the sample Java source code, HTML, JavaServer Pages files, servlet classes, and enterprise beans in the following directories on iSeries:

```
/QIBM/ProdData/WebAS5/Base/samples/src/samplename
```

For example, the source code for the PlantsByWebSphere sample is located in this directory:

```
/QIBM/ProdData/WebAS5/base/samples/src/PlantsByWebSphere
```

The subdirectory that contains source for a Web Module ends with “WAR” and those for a corresponding EJB module ends with “EJB”. All the Java source is in subdirectories according to its package name. For example, if a package name is `com.ibm.somepackage`, the source is in the `com/ibm/somepackage` subdirectory.

You may find it helpful to treat the samples gallery as a starting point for to further your understanding of WebSphere Application Server topics. You may want to change the samples source code for experimentation and testing.

Note: It is recommended that, before making changes, you copy the source files from the `/QIBM/ProdData/WebAS5/Base/samples/src` directory to your own directory.

To access the samples in the WebSphere Application Server samples gallery that are included with the WebSphere Application Server on your iSeries server, follow these steps:

1. Verify that Option 3, the WebSphere Application Server samples, is installed, by performing these steps:
   a. From your iSeries command prompt, enter the following command:

      ```
      DSPSFWRSC
      ```

      If you see the following entry, the samples gallery is installed:

      ```
      5733WS5 3 5050 WAS V5.0 Samples
      ```

   b. If Option **3**, the WebSphere Application Server samples, is not installed, install it (see 2.11.3, “Parameters for SETUP script and RUNJAVA command” on page 28).

2. Use the `installsamples` script to install the samples into a WebSphere Application Server instance. This script is located in IFS in the `/QIBM/ProdData/WebAS5/Base/samples/bin` directory. Follow these steps to install the samples:

   a. Start the instance and application server into which you want to install the samples; see 6.5.3, “Starting a specific application server” on page 117.

   b. From the OS/400 command line, start a QShell session:

      ```
      STRQSH
      ```

   c. Invoke the `installsamples` script:

      ```
      /QIBM/ProdData/WebAS5/Base/samples/bin/installsamples -instance instanceName
      -server serverName
      ```

      Here, `instanceName` is the name of the WAS instance, and `serverName` is the name of the application server in this instance into which you want to install the samples. If the `-instance` parameter is not specified, the default instance is used.

      You can see the installed applications in the administrative console, as shown in Figure 6-81.

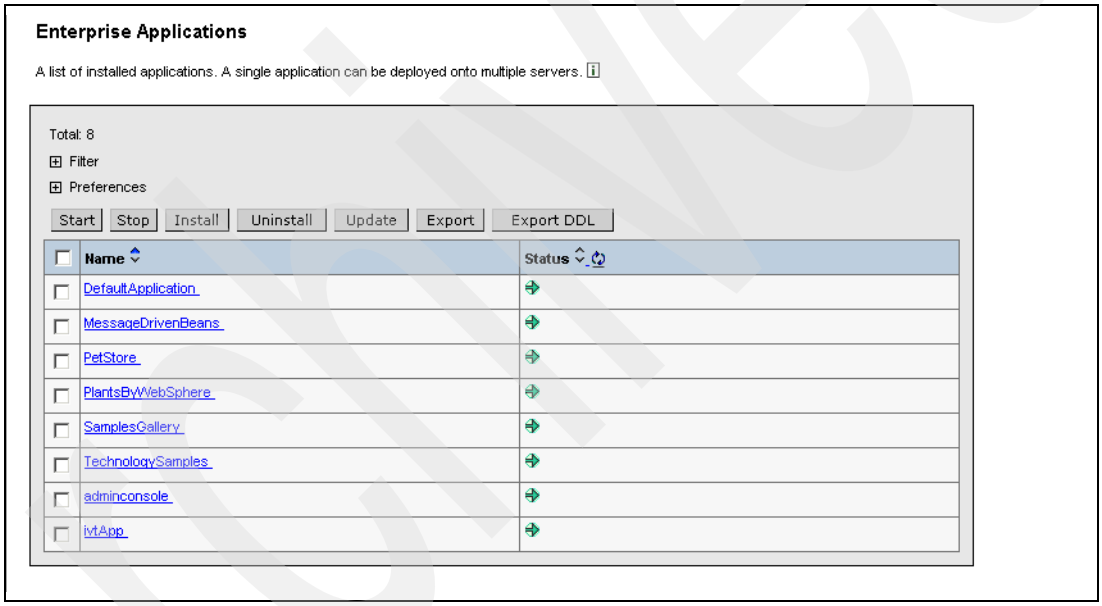


Figure 6-81 installed applications on our sample instance

3. Configure and start an HTTP server instance for your WAS instance (if you have not already done this before); see 6.6, “Configuring an HTTP server instance” on page 132.
4. Configure the virtual host; see 6.7.2, “Configuring a virtual host” on page 161.
5. Update the Web server plug-in configuration; see 6.7.4, “Updating Web server plug-in configuration” on page 171.
6. Restart the WAS server; see 6.8, “Restarting the WebSphere Application Server” on page 176.
7. To invoke the WebSphere Application Server samples’ main page, point a browser to the following URL:

   ```
   http://your.server.name:port/WSsamples/index.jsp
   ```

   Here, `your.server.name` is the host name of your iSeries server and `port` is either your internal HTTP port number or your external HTTP port number. 9080 is the default internal HTTP port number.

6.13 Working with Web applications in a WAS environment

In this section, we explore the management functions for working with Web applications in the WAS environment. We use three different types of applications to do this:

- A JDBC application (see 6.13.2, “Working with the JDBC applications” on page 189)
- A JMS application (see 6.13.3, “JMS Administration in WebSphere 5.0” on page 220)
- A WebFaced application (see 6.13.6, “Installing a sample WebFacing application” on page 245)

6.13.1 Administering JDBC providers and DataSources

WebSphere Application Server supports all of the resources defined by the Java 2 Platform, Enterprise Edition (J2EE). In this section, we describe the configuration of JDBC DataSources to offer support so that the applications can access a database via the JDBC providers. A JDBC provider represents a JDBC driver to access the database. DataSources are assigned to a JDBC provider in WAS version 5.0.

A DataSource object represents the logical name of a JDBC-enabled database. This shields the enterprise bean developer from the underlying physical location of the database.

In the first part of this section we describe the WAS environment we used — an additional WAS instance — where we configure the JDBC provider and the DataSource. Later on we install our sample application MYBankCMP, which uses these JDBC elements to make connections to the DB2/400 UDB database. You can also use the WAS default instance to set up this configuration.

After finishing the installation, we show how we tested the application through a Web interface that uses the local interfaces to create accounts and transfer funds.

WebSphere 5.0 types of DataSources

WebSphere 5.0 offers two different types of DataSources:

- The V4.0 DataSources are a carry-over from the previous releases and only support EJB 1.1 persistence. They are available for compatibility reasons. They can be used with existing EJBs, so that you can run them as they are, with no changes or need for redeployment.
- The V5.0 DataSources are the ones that utilize the Relational Resource Adapter. It supports the EJB 2.0 CMP persistence model. They also support for EJB 1.1 CMP persistence. However, 1.1 persistence can be used only with EJB modules that have a EJB 2.0 deployment descriptor. You can include EJBs with EJB 1.1 persistence in an EJB 2.0 module (the specifications support both persistence models). These modules can use either V4.0 or V5.0 DataSources — V5.0 is recommended.

Figure 6-82 outlines the possible combinations of EJB modules and DataSources:

- A J2EE 1.2 application module (with EJB 1.1 persistence) needs to use the “old” WebSphere 4.0 DataSources (in other words, applications carried over “as-is” from WebSphere 4.0 will use these DataSources).
- A J2EE 1.3 application may contain both EJB 1.1 and EJB 2.0.
- For EJB 2.0 persistence, there is only one choice — the new DataSources.
- For EJB 1.1 persistence inside a J2EE 1.3 module, you may choose between the V4.0 and the V5.0 DataSources — although the latter are recommended (for performance, caching, etc.)

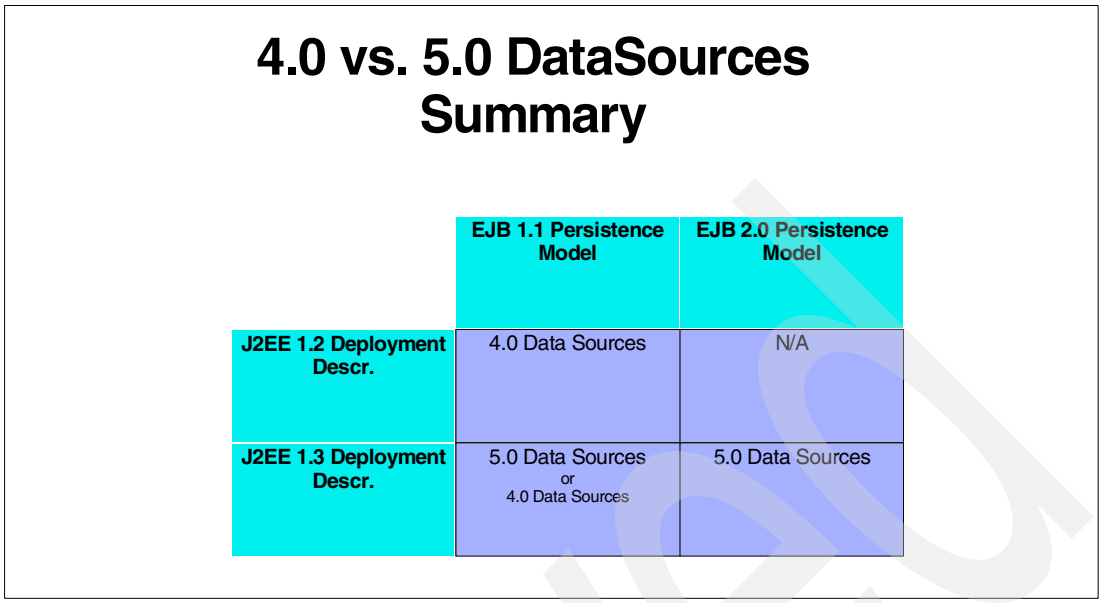


Figure 6-82 EJB modules and DataSources

In order to be used by the CMP entity beans, V5.0 DataSources must be associated with a JCA Connection Factory. The JNDI name of the connection factory is derived from the name of the DataSource by prepending “eis/” and appending “_CMP” (see Figure 6-83).

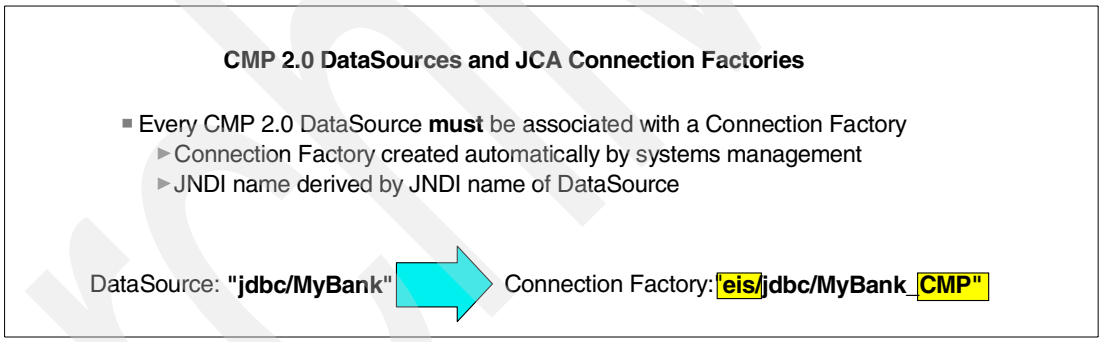


Figure 6-83 V5.0 DataSources and JCA Connection Factories

JDBC drivers

When your applications access DB2 Universal Database (UDB) for iSeries, you have to choose which JDBC driver type you use. There are JDBC drivers from the IBM Developer Kit for Java (native driver) and IBM Toolbox for Java JDBC drivers. Also, there are JDBC drivers which support JTA, or do not. Your choice of which JDBC driver to use depends on the following requirements:

- Local or remote database access: You can configure WebSphere Application Server for iSeries to access local or remote databases.
- Java Transaction API (JTA): If you need two-phase commit capability, you need to use a JTA-enabled driver. Enterprise JavaBeans that access multiple database tables within a transaction boundary require JTA support.
- The OS/400 Version your iSeries database server runs on: For the native drivers, you have to choose the according driver depending the OS/400 version.

Table 6-4 summarizes when to use each JDBC driver.

Table 6-4 iSeries JDBC drivers

Driver name in WAS		Local database access	Remote database access	JTA enabled	OS/400 Version
DB2 UDB for iSeries (Native - V5R2 and later)	IBM Developer Kit for Java JDBC Driver (native Driver)	X			V5R2 and later
DB2 UDB for iSeries (Native XA - V5R2 and later)	IBM Developer Kit for Java JDBC Driver (native Driver)	X		X	V5R2 and later
DB2 UDB for iSeries (Native - V5R1 and earlier)	IBM Developer Kit for Java JDBC Driver (native Driver)	X			V5R1 and earlier
DB2 UDB for iSeries (Native XA - V5R1 and earlier)	IBM Developer Kit for Java JDBC Driver (native Driver)	X		X	V5R1 and earlier
DB2 UDB for iSeries (Toolbox)	IBM Toolbox for Java JDBC driver		X		
DB2 UDB for iSeries (Toolbox XA)	IBM Toolbox for Java JDBC driver		X	X	

When running under WebSphere Application Server on the iSeries server, the native driver is the preferred driver. The IBM Toolbox for Java JDBC driver can be used to connect to local iSeries databases; however, you can expect better performance by using the native JDBC driver for local database connections.

6.13.2 Working with the JDBC applications

To better understand the steps involved in the process of deploying a JDBC application to WAS, we guide you through the deploying process for a simple bank application, `MyBankCMP`.

Introducing of the MyBankCMP sample application

We use the MyBankCMP application to demonstrate the process of installing and deploying a Web application in WebSphere Application Server V5.0 for iSeries.

The MyBankCMP application has an entity bean which represents an Account object. WebSphere will actually handle the persisting of the account information to a database. We need to configure a JDBC Provider and a DataSource for the MyBankCMP application to use for persisting data. The application features Container-Managed Persistence (CMP) and Local Interfaces, both which are part of the J2EE 1.3 specification.

Figure 6-84 represents the design of the application.

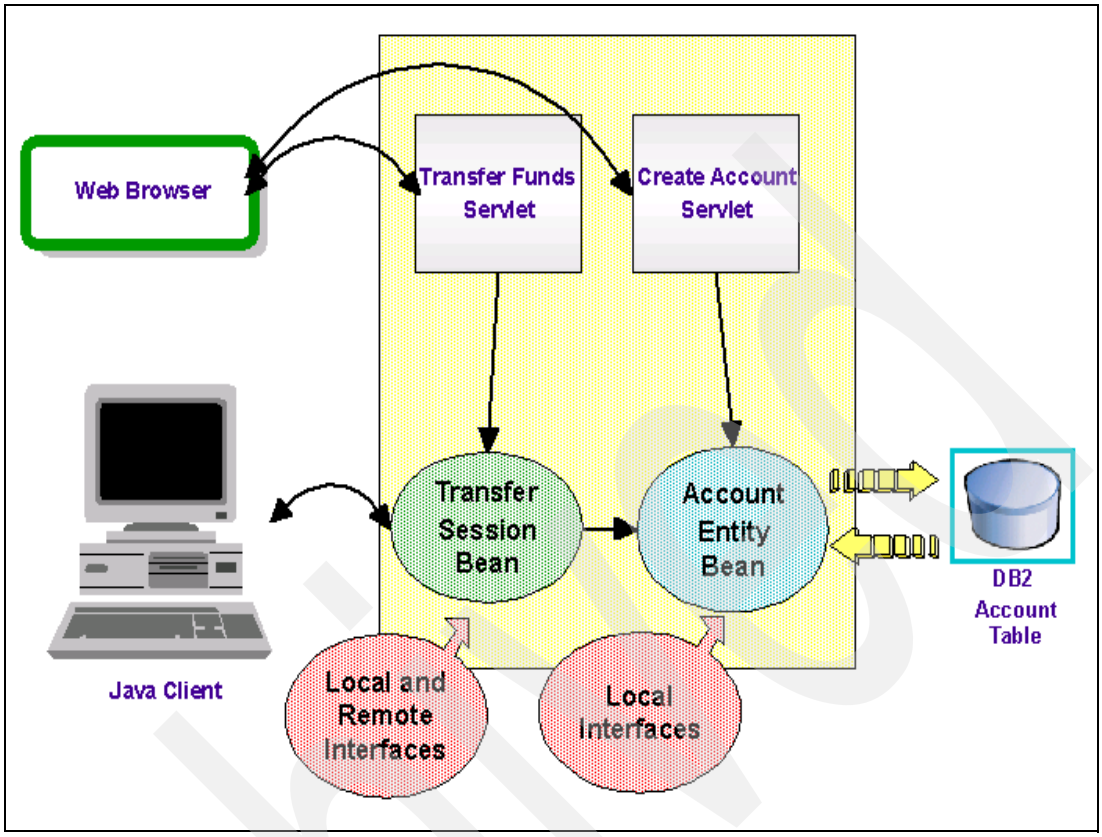


Figure 6-84 Overview of the MyBankCMP sample application

Setup steps to install and run the My Bank sample application

We shall follow these steps to install and deploy the application:

1. First, create a new WAS instance with the `crtwasinst` script, in which our MyBankCMP application will be installed and run. For more information, see “Creating a new WAS instance” on page 113.

   Use the following parameters to create the instance (see Figure 6-85):

   ```
   crtwasinst -instance bankap -exthttp 10200 -inthttp 10201 -admin 10202 -portblock
   10203 -nodefaultapps
   ```

```
crtwasinst -instance bankap -exthttp 10200 -inthttp 10201 -admin 10202 -portblock 10203
-nodefaultapps
Creating instance bankap...
ADCP0005I: Using cell RCHAS07_bankap, node RCHAS07_bankap and server bankap.
ADCP0006I: Embedded JMS enabled.
Instance bankap created.
Instance root directory is /QIBM/UserData/WebAS5/Base/bankap.
The application server name is bankap.
 Ports:
   External HTTP:  10200
   External HTTPS: 443
   Name service:   10203
   JMS secure:     10204
   JMS queued:     10205
   JMS direct:     10206
   DRS client:     10207
   SOAP:           10208
   SAS:            10209
   CSIV2 Mutual:   10210
   CSIV2 Server:   10211
   Internal HTTP:  10201
   Admin:          10202
   Admin SSL:      10212
 $
```

Figure 6-85 create instance bankap

2. Start the WAS instance `bankap` with the `startServer` script (see Figure 6-86), for more information, see 6.5.3, “Starting a specific application server” on page 117.

```
startServer bankap -instance bankap
CPC1221: Job 034774/QEJBSVR/BANKAP submitted to job queue QEJBJOBQ in
library QEJBAS5.
EJB6123: Application server started.
 Cause . . . . . :   Application server bankap in Base instance bankap has
   started and is ready to accept connections on admin port 10202.
$
```

Figure 6-86 The startServer bankap

3. Create an HTTP server for the WAS instance bankap and start it. For more information how to do that, see 6.6, “Configuring an HTTP server instance” on page 132.
4. Create an iSeries user profile WSDEMO with password WSDEMO1.

   This user profile is used to access to the database via a DataSource.
5. Set up the collection `bankdata`; see “Setting up the bankdata database collection” on page 192.
6. Create the JDBC provider and DataSource; see “Configuring JDBC Providers and DataSource” on page 195.
7. Assign the above user profile (see step 16 on page 202) to the DataSource.
8. Restart the WAS instance bankap; see 6.8, “Restarting the WebSphere Application Server” on page 176.
9. Install the MyBankCMP application; see “Installing MyBankCMP sample application” on page 206.

10. Update the Web server plugin; see “Updating Web server plug-in configuration” on page 171.
11. Test the MyBankCMP application; see “Running the new MDB application” on page 242.

Setting up the bankdata database collection

First of all, we need to set up the collection that MyBank sample application will use. These are the steps:

1. Start the iSeries Navigator.
2. Connect to the iSeries system on which you’re going to run the sample application.
3. In the iSeries Navigator, expand **<iSeries system> -> Database -> <DB name>**.
4. Right-click **Libraries** and select **New Library** (see Figure 6-87).

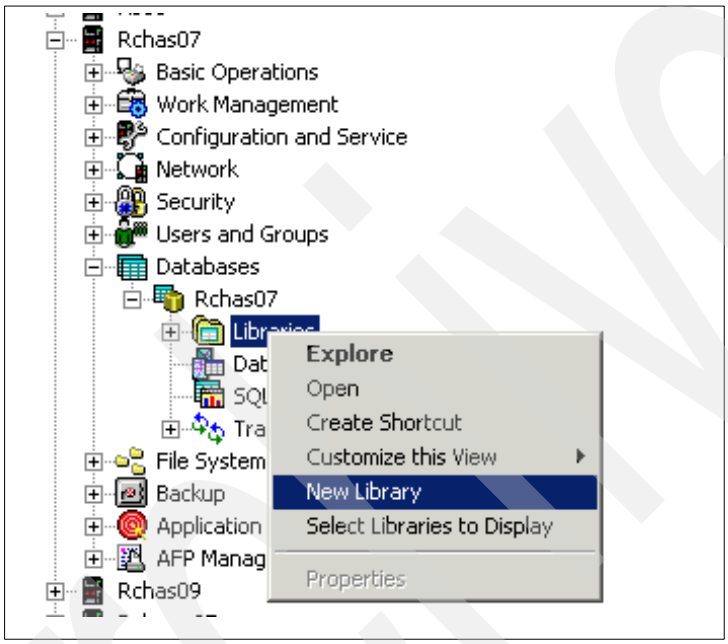


Figure 6-87 Creating a new library

5. In the pop-up window, type `bankdata` as the name of the new library.
6. Click **OK**. Now the BANKDATA library has been created and added to the list of displayed libraries.
7. Right-click the relational database (Rchas07 in our example; see Figure 6-87) and select **Run SQL Scripts**.
8. Open the table.ddl file in the `MyBankCMP application` folder (see Appendix B, “Additional material” on page 551).
9. Copy two SQL statements from this file.
10. Paste them into the Run SQL Scripts window.
11. Click the **Run All** button (see Figure 6-88).

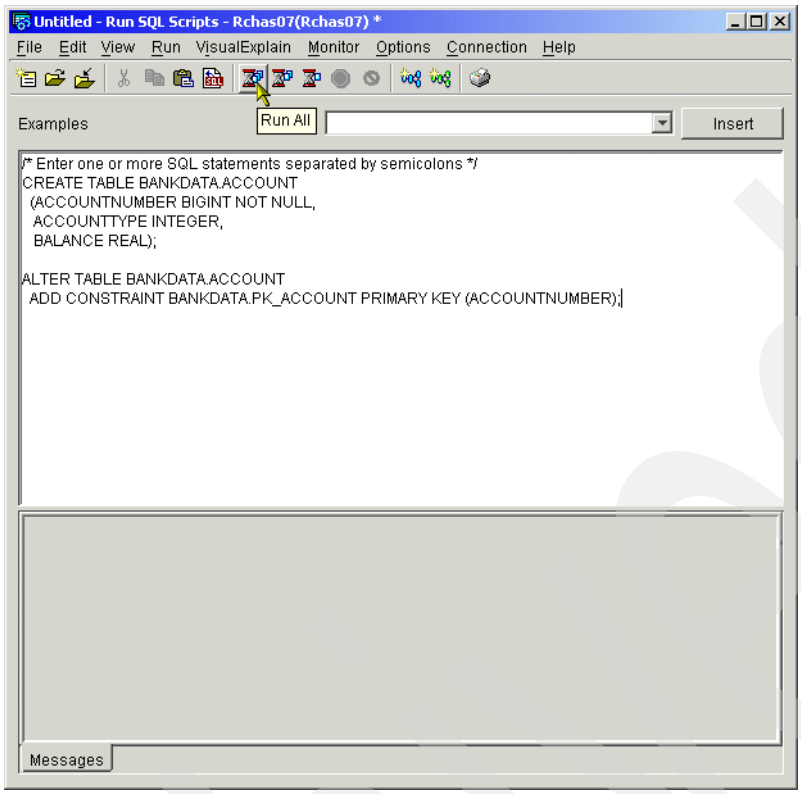


Figure 6-88 Running the SQL statements

12. Close the Run SQL Scripts window.
13. Click **No** in the pop-up window that asks if you want to save changes.
14. Right-click **BANKDATA** in the iSeries Navigator and select **New -> Journal** (see Figure 6-89). A database table has to be journaled in order to be used by EJB.

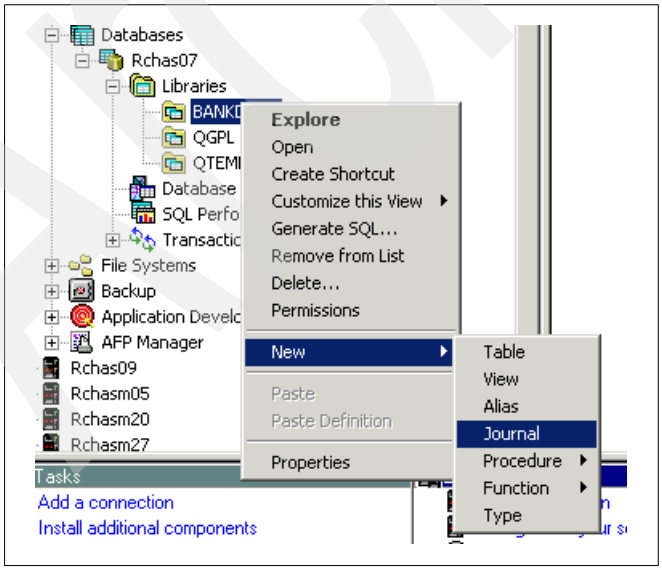


Figure 6-89 Creating a Journal

15. Fill in the name of the journal and library to hold a receiver (see Figure 6-90).

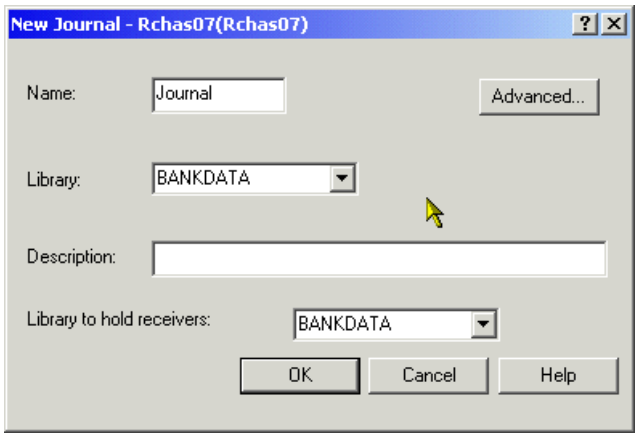


Figure 6-90 Journal parameters

16. Click **OK**.
17. Click **BANKDATA** in the iSeries Navigator.
18. In the right-hand pane right-click the journal name and select **Starts and ends table journaling**.
19. In the new pop-up window, expand **BANKDATA** and select **ACCOUNT**.
20. Click the **Add** button (see Figure 6-91).

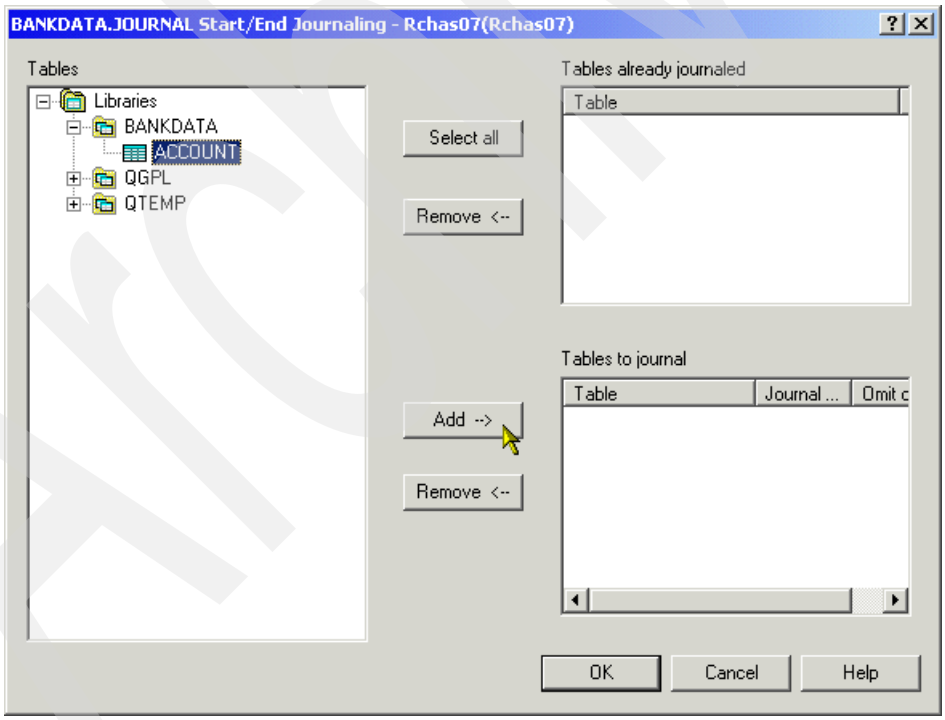


Figure 6-91 Start journaling the ACCOUNT table

21. Click **OK**.

Configuring JDBC Providers and DataSource

Use the administrative console for creating the JDBC Providers and DataSource:

1. Make sure that your instance is started and running. If it is not started, use the `startServer` script, for more information, see 6.5.3, “Starting a specific application server” on page 117.

   ```
   startServer bankap -instance bankap
   ```

2. Access your administrative console for your instance; see 6.7, “Working with the Administrative Console” on page 159.

   Because our bankap instance has the admin port 10202 defined (see Figure 6-85 on page 191), we use:

   ```
   http://rchas07:10202/admin
   ```

3. JDBC Providers, DataSources, and other resources are organized under the Resources heading. Expand **Resources** and click **JDBC Providers** to configure the DataSource which your application will use to persist data; see Figure 6-92.

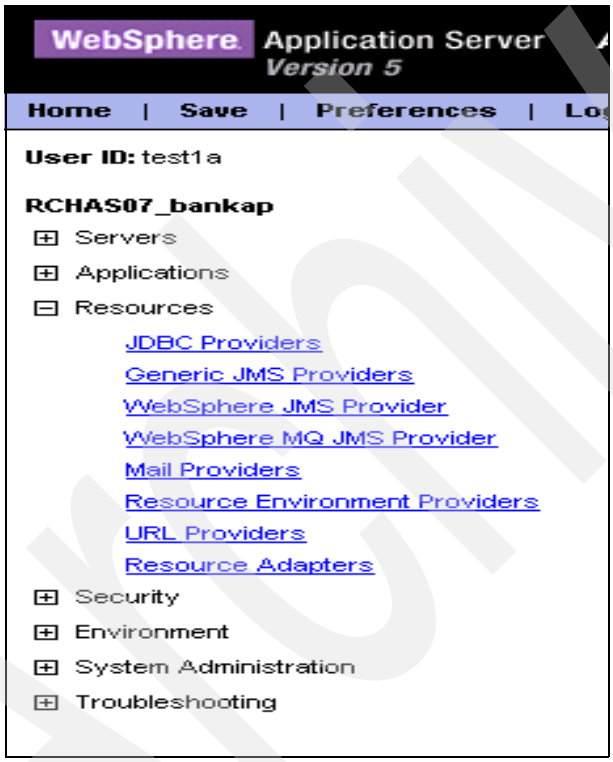


Figure 6-92 Resources - JDBC Providers

4. The existing JDBC Providers will be displayed. Verify that the **Scope** is set to **Node**, and then click **New** to create a DB2 Provider; see Figure 6-93.

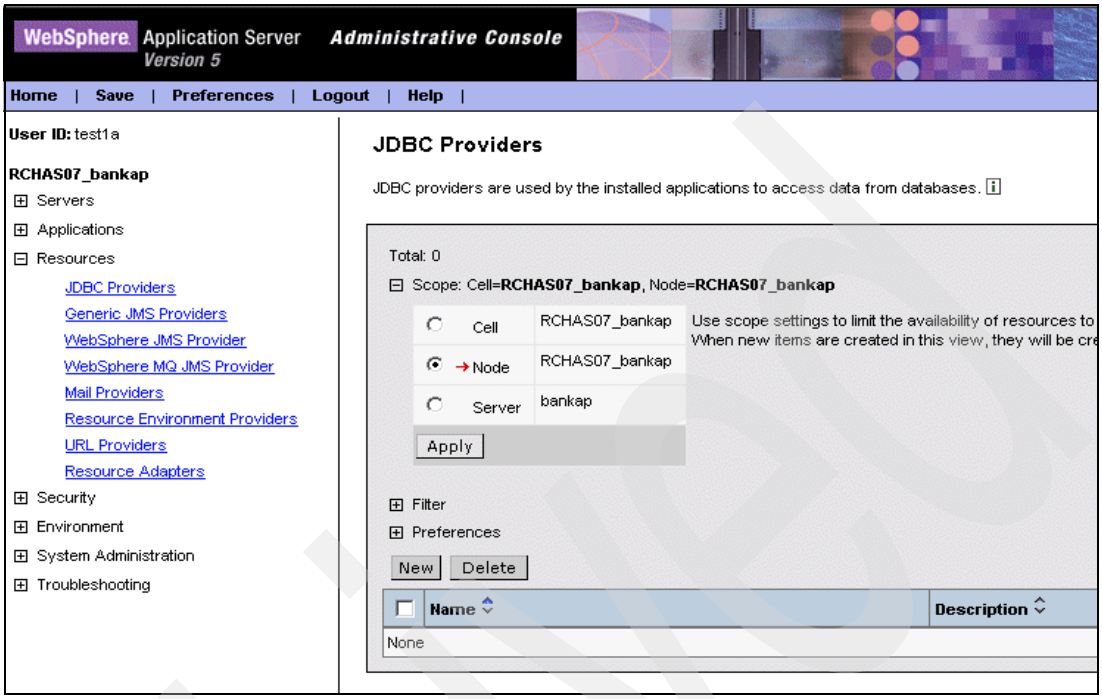


Figure 6-93 JDBC Provider new

5. Select the DB2 JDBC Provider according to your needs; see “JDBC drivers” on page 188.

   For our MyBankCMP sample application, we choose the **DB2 UDB for iSeries (Native XA - V5R2 and later)**.

6. Click **Apply**; see Figure 6-94.

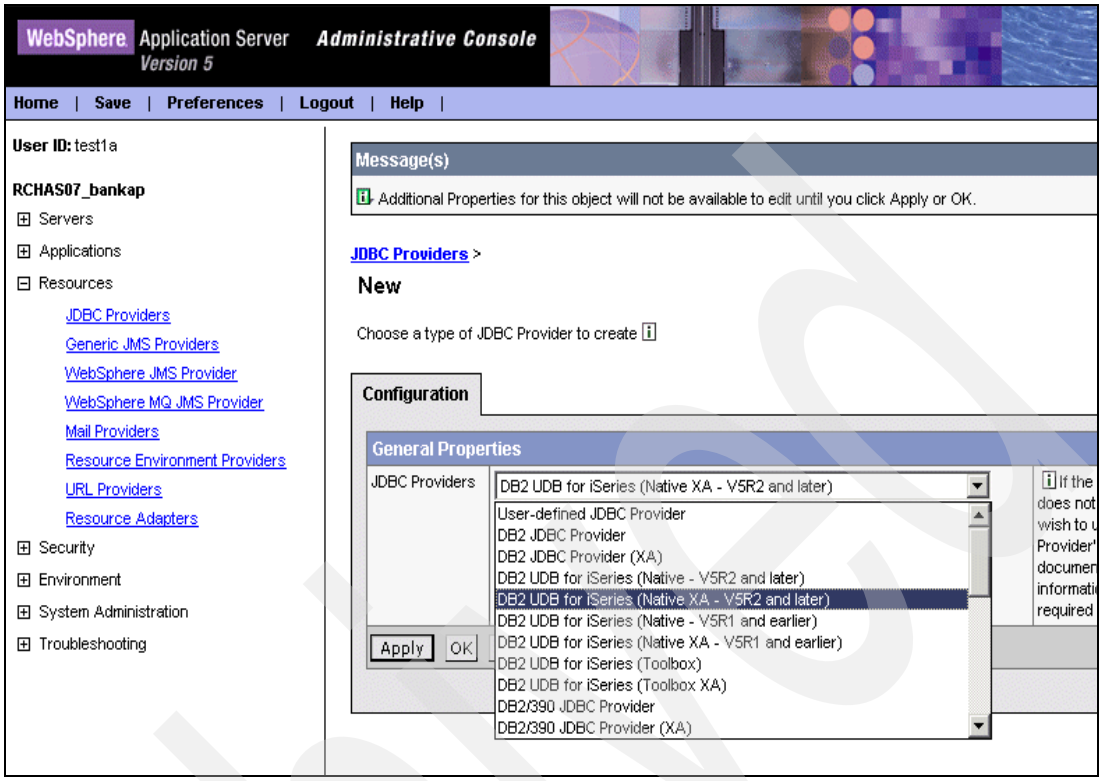


Figure 6-94 Select DB2 JDBC Driver

7. The configuration properties for the JDBC Provider will be displayed. They are all correct, so no changes have to be done here. Click **OK**; see Figure 6-95 and Figure 6-96.

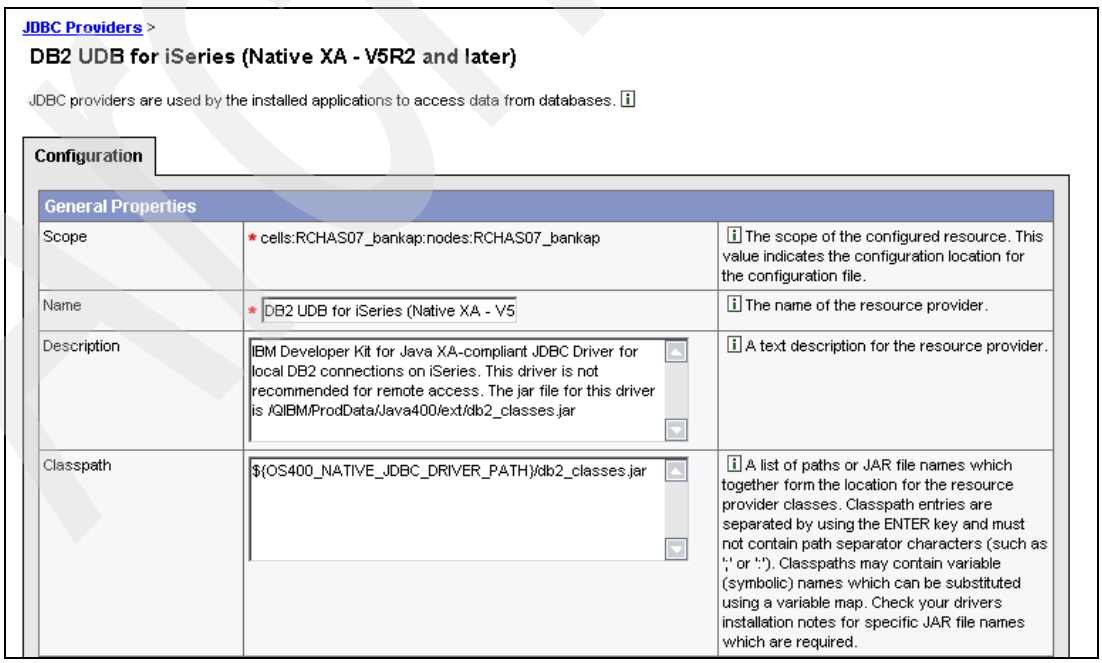


Figure 6-95 Configuration properties for the JDBC Provider - Part 1

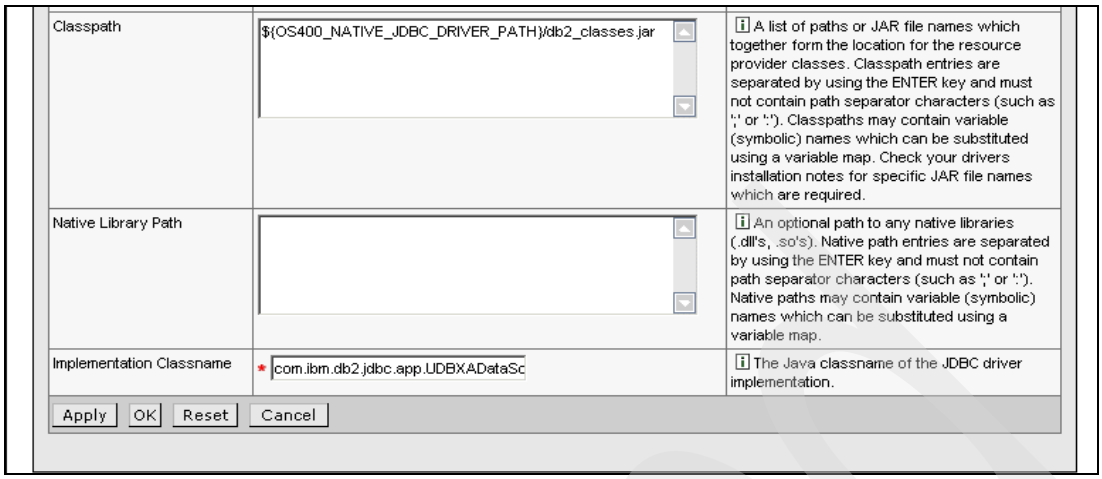


Figure 6-96 Configuration properties for the JDBC Provider - Part 2

8. In the DB2 JDBC Provider created, you will now create a DataSource. Click the DB2 JDBC Provider — in our case, **DB2 UDB for iSeries (Native XA - V5R2 and later)**; see Figure 6-97.

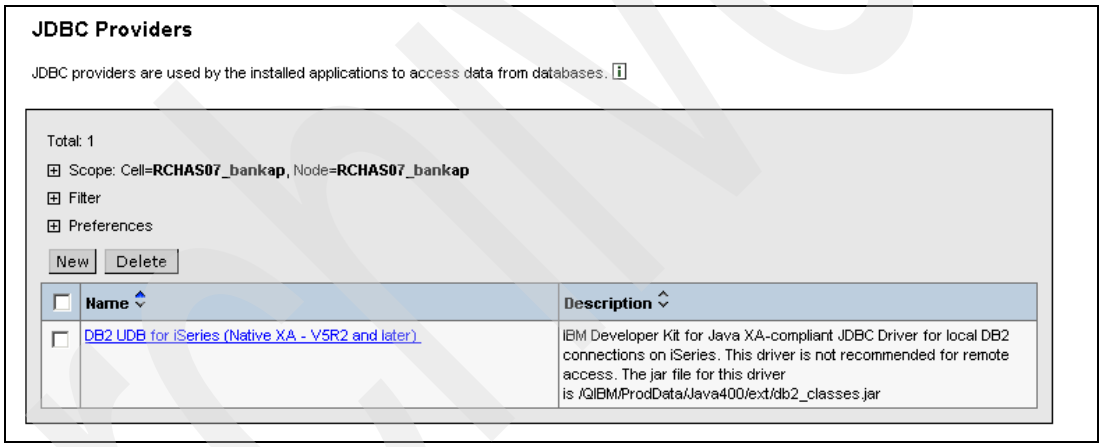


Figure 6-97 JDBC Provider created

9. The configuration properties for the JDBC Provider will be displayed. Click **Data Sources** (you may have to scroll down); see Figure 6-98.

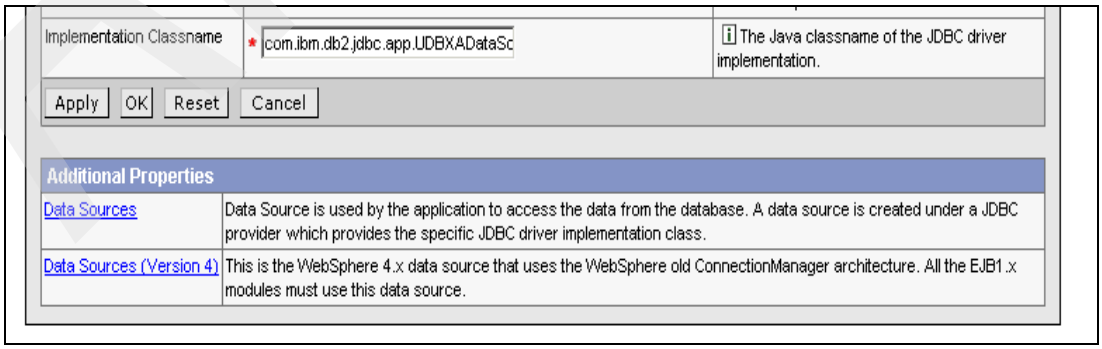


Figure 6-98 Create DataSource

10. The list of Data Sources for the DB2 JDBC provider will appear. Click **New** to create a DataSource; see Figure 6-99.

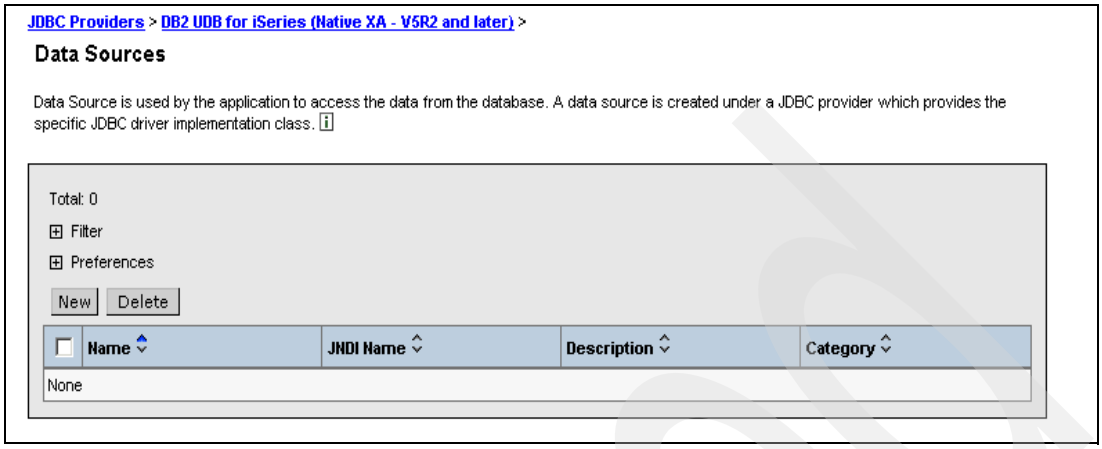


Figure 6-99 Create new DataSource

11. The empty General Properties for the DataSource (see Figure 6-100) will be displayed. Enter the following:

    Name: **MyBankDS**

    This is the name of the DataSource, and you can use any name.

    JNDI name: **jdbc/MyBank**

    In our case, we select the checkbox next to **Use this DataSource in container managed persistence CMP**.

    It specifies if this DataSource will be used for container managed persistence of EJBs. This will cause a corresponding CMP connection factory to be created. So, whether to select or not depends on your application which uses this DataSource.

    Description: **MyBank Datasource**

12. Click **OK** (you may have to scroll down).

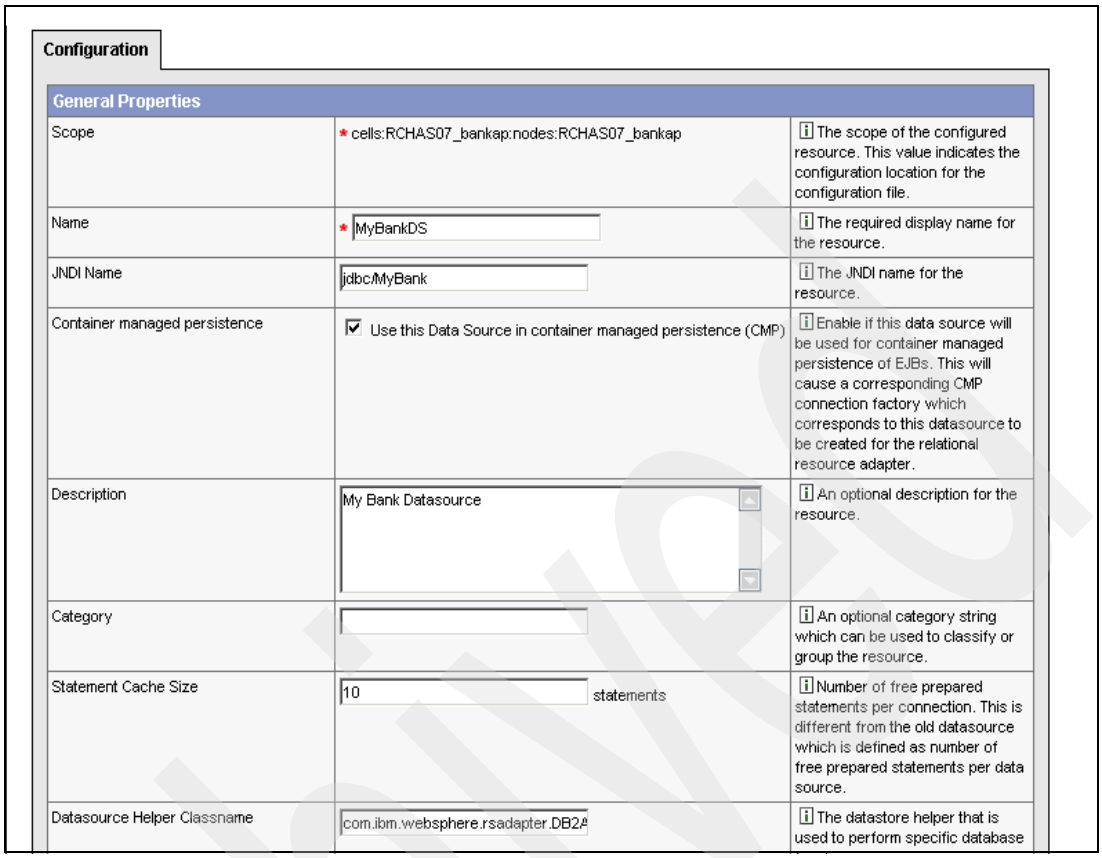


Figure 6-100 General Properties for the DataSource

Note: Ensure that the box is checked for "Use the data source in container managed persistence (CMP)" when your application works with CMP EJBs. If this box is not checked, you will have problems at runtime (the correspondent connection factory is not going to be created).

13. The DataSource is now created, however, there are a number of additional configuration properties which still need to be set. Click **MyBankDS**; see Figure 6-101.

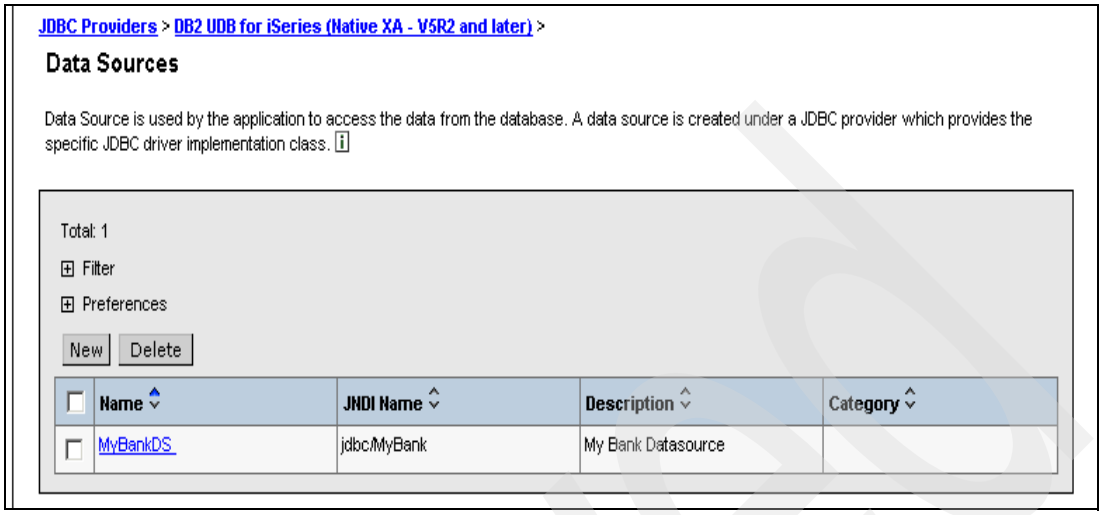


Figure 6-101 Update additional properties

14. The DataSource configuration properties will be listed again. Click **J2C Authentication Data Entries** (you will have to scroll down, it is in the **Related Items** section); see Figure 6-102.

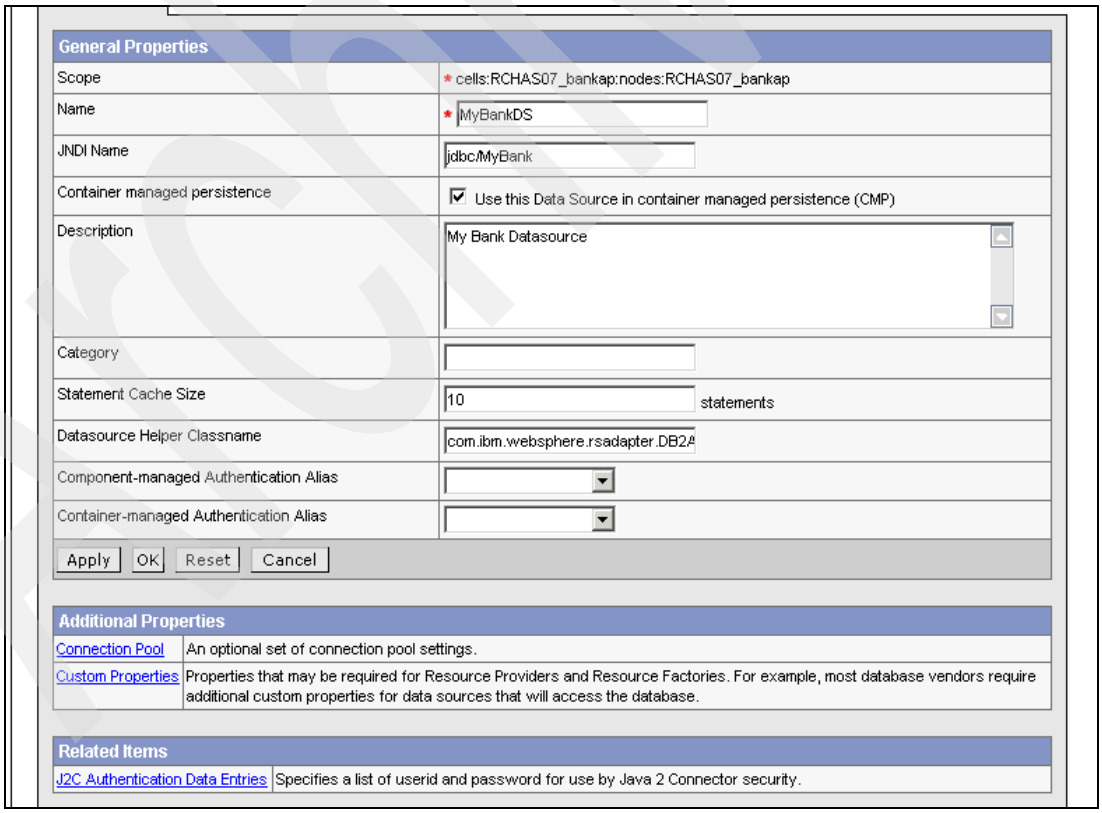


Figure 6-102 J2C Authentication Data Entries Part 1

15. As shown in Figure 6-103, the J2C Authentication Data Entries panel appears. Click **New** to create a new J2C Authentication Data Entry.

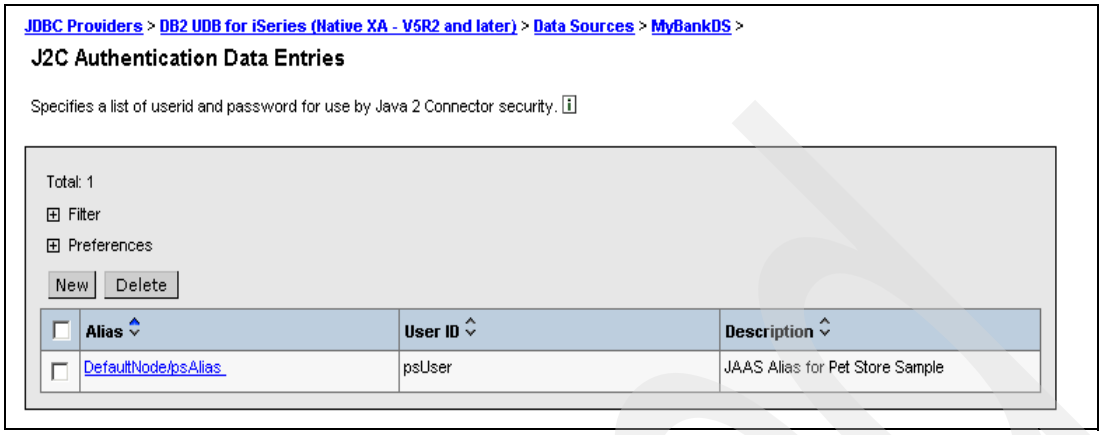


Figure 6-103 J2C Authentication Data Entries panel

16. Enter an Alias of **MyBankAlias**, a USER ID of **wsdemo**, and a password of **wsdemo1**. Click **OK**; see Figure 6-104.

    You can use any name for the Alias.

> **Note:** The user profile and password defined here must exist on your iSeries server. All connections obtained from this DataSource use the user ID and password specified here.

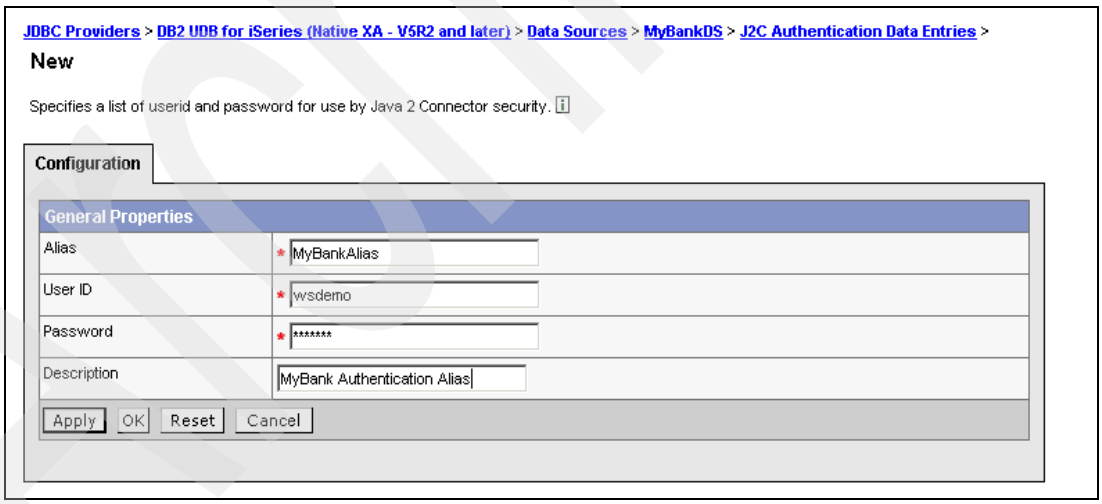


Figure 6-104 Define userid and password for use by Java 2 Connector security

17. Take a shortcut back to the DataSource — click **MyBankDS** in the right upper link list; see Figure 6-105.

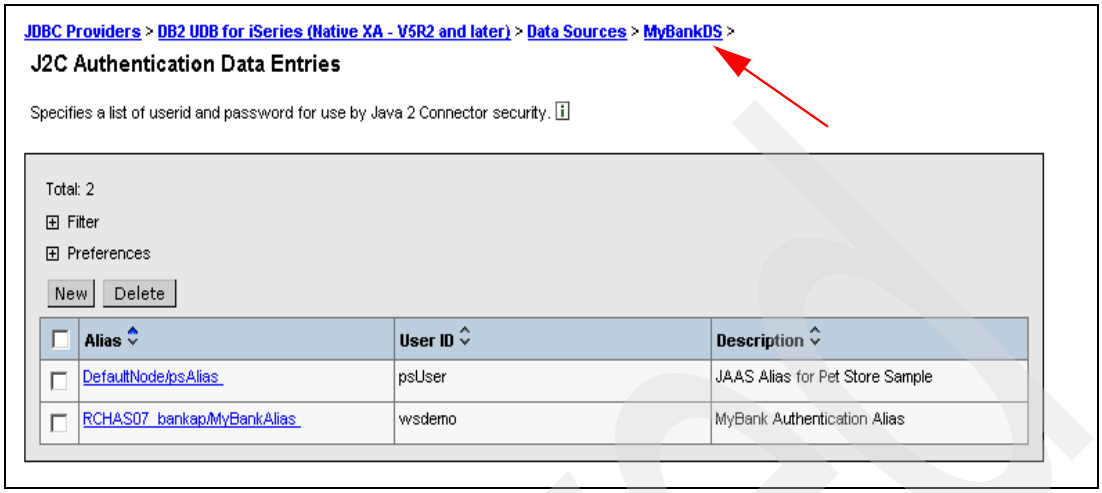


Figure 6-105 Take a shortcut back to MyBankDS

18. Scroll down and click **Custom Properties**; see Figure 6-106.

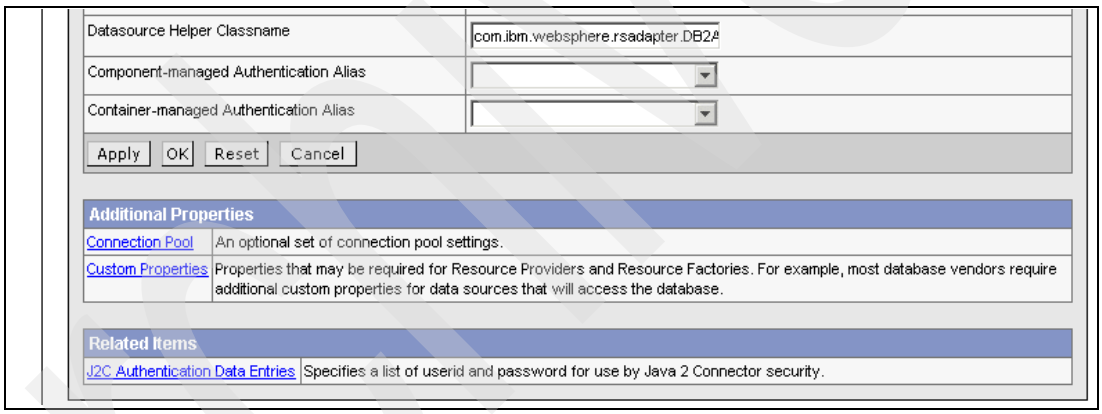


Figure 6-106 Select Custom Properties

19. In the next panel, scroll down and click **databaseName**; see Figure 6-107.

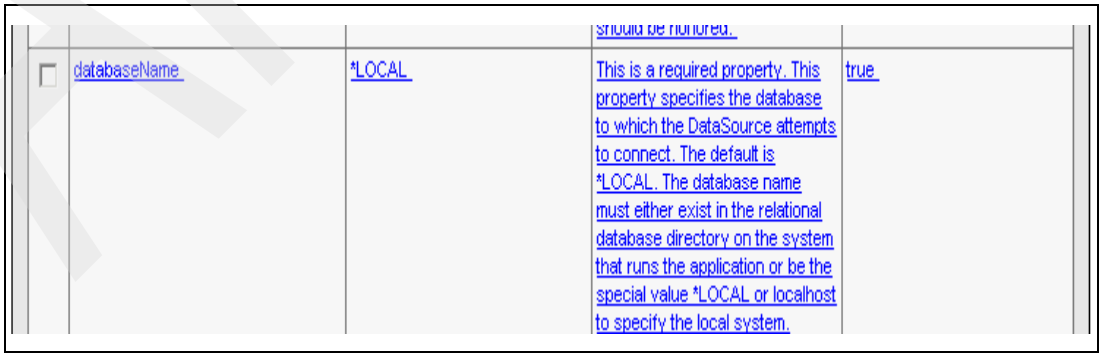


Figure 6-107 Select database name

20. Here, you set the name of the database (collection) your application will access.

    The format of this value is '*LOCAL/CollectionName', where '*LOCAL' represents the name or the database to which the DataSource attempts to connect and 'CollectionName' represents the name of the of the collection (library).

    *LOCAL represents a database that is defined as *LOCAL on the Work with Relational Database Directory Entries (WRKRDBDIRE) command (see Figure 6-108). In our case, the database name is RCHASE5C.

    For our sample, we define database as ***LOCAL/bankdata**.

    Note: When the database with which your application works is on the same iSeries as WAS, you can use *local here; otherwise:

    - For the native JDBC driver: It is the name of the database you can see with the iSeries command WRKRDBDIRE for the remote database.
    - For the IBM Toolbox for Java JDBC driver: It is the host name of the remote iSeries.

```
               Work with Relational Database Directory Entries

Position to  . . . . . .

Type options, press Enter.
  1=Add   2=Change   4=Remove   5=Display details   6=Print details

        Relational          Remote
Option  Database            Location                Text

        MYDB                LOOPBACK                Entry added by system
        MYDB2               LOOPBACK                Entry added by system
        DBCOOK              9.5.92.31
        DBEUOPS             9.5.92.31
        RCHASE5C            *LOCAL                  Entry added by system




                                                                      Bottom
F3=Exit   F5=Refresh   F6=Print list   F12=Cancel
```

Figure 6-108 WRKRDBDIRE

21. Click **OK**; see Figure 6-109.

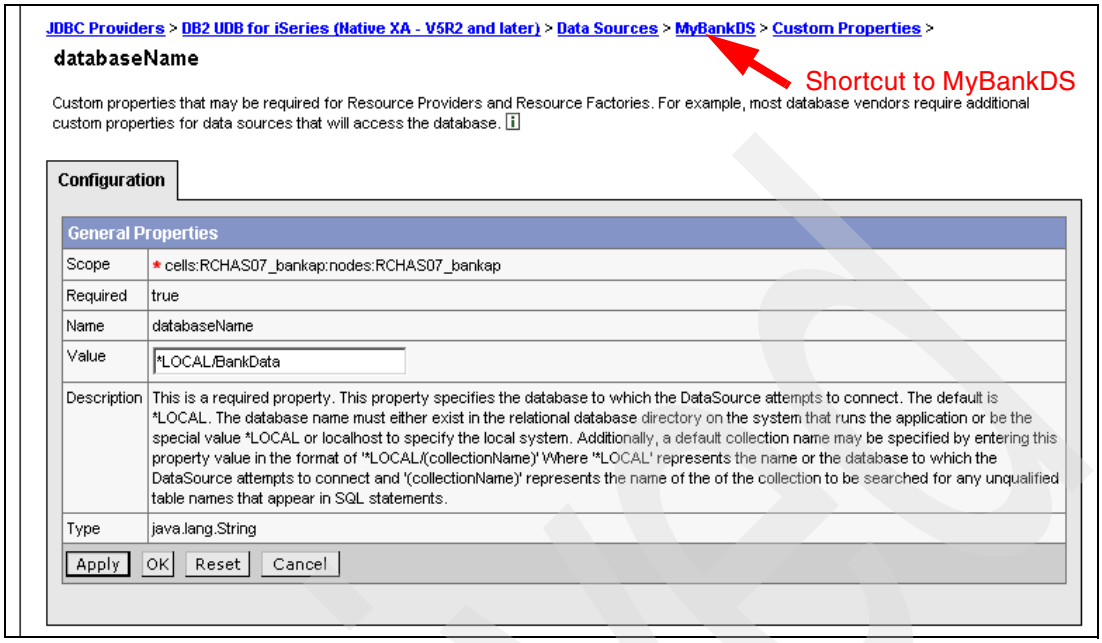


Figure 6-109 Define database name and collection name

22. Take the shortcut back to **MyBankDS** (see Figure 6-109).

23. Assign the new J2C Authentication Data Entry to the **Component-managed Authentication Alias** and also to the **Container-managed Authentication Alias** in the General Properties section (you may have to scroll down):

   – From the **Component Managed Authentication Alias** drop-down, select the newly created MyBankAlias and click **Apply**.

   – From the **Container-managed Authentication Alias** drop-down, select the newly created MyBankAlias and click **OK**; see Figure 6-110.

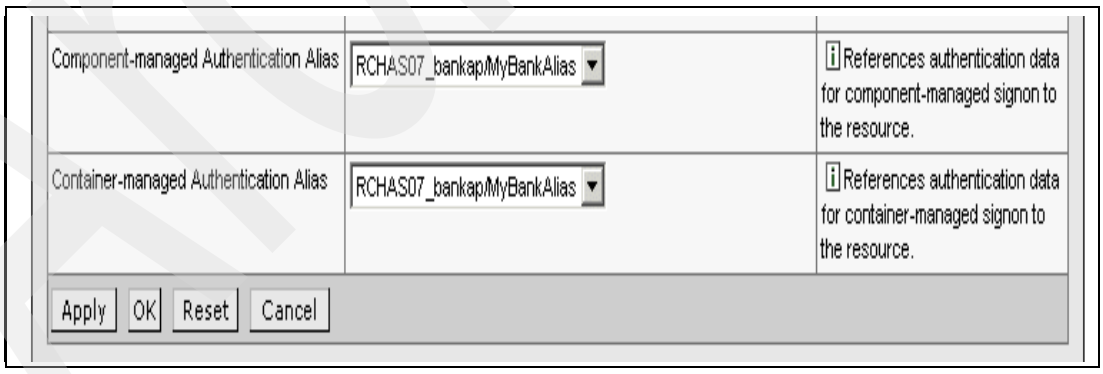


Figure 6-110 Assign MyBankAlias to DataSource

24. Your DataSource is now completely defined. Your configuration changes to your server have to be saved. Click **Save** to apply changes to the master configuration.

25. **Save** your workspace changes to the master configuration.

26. Before this new JDBC Resource and DataSource can be used by an application, the server will need to be restarted. To do this, refer to 6.8, “Restarting the WebSphere Application Server” on page 176.

Installing MyBankCMP sample application

During installation you can specify binding information for the application running on your WebSphere Server. The application which you will install is a small banking application. The application features Container-Managed Persistence (CMP) and Local Interfaces, both which are part of the J2EE 1.3 specification. After you have finished installation, you will test the application through a Web interface which uses the local interfaces to create accounts and transfer funds.

Use the administrative console for installing applications. To install the MyBankCMP application, follow these steps:

1. Be sure that your instance is started and running; see 6.5.4, “Verifying that the WAS environment has started” on page 119.
2. Access your administrative console for your instance; see 6.7, “Working with the Administrative Console” on page 159.
3. Expand **Applications** and click **Install New Application**; see Figure 6-111.

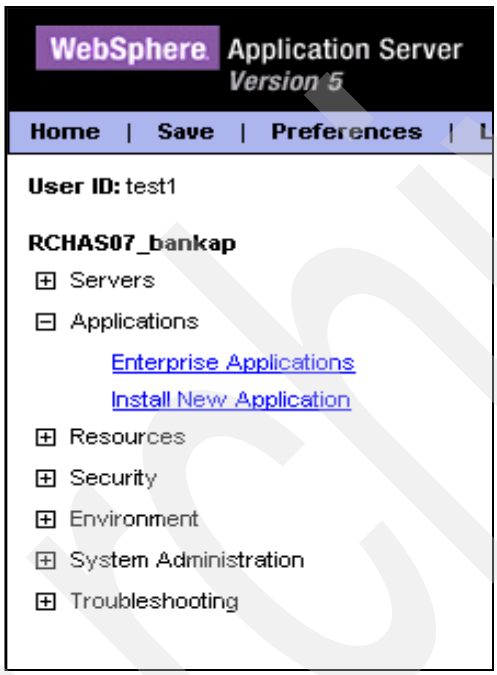


Figure 6-111 Install applications start

4. In the Preparing for the application installation panel select **Local path** radio button and click **Browse**.

   Navigate through the directory structure and select the .EAR file that includes the application you want to install (in our case, mybankcmp.ear) and click **Open**; see Figure 6-112.

> **Note:** Refer to Appendix B, “Additional material” on page 551 for the instructions for downloading the application.

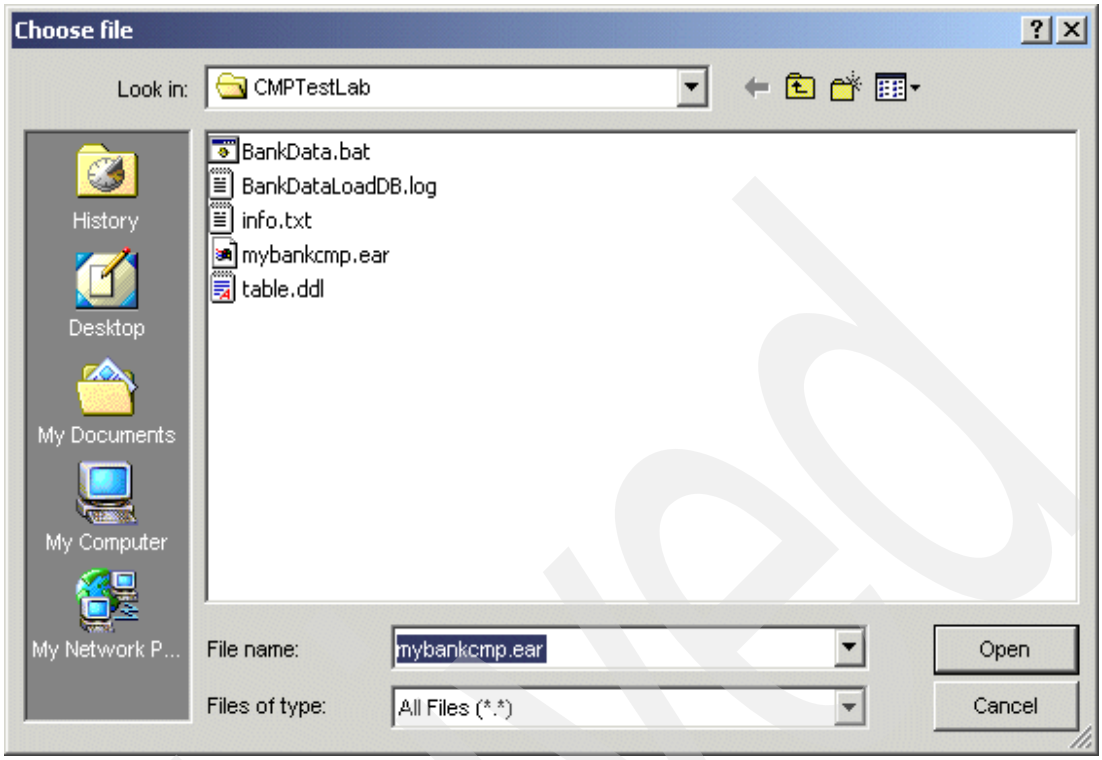


Figure 6-112 select myBankCMP.ear

5. Back in the Preparing for the application installation panel, click **Next**; see Figure 6-113.

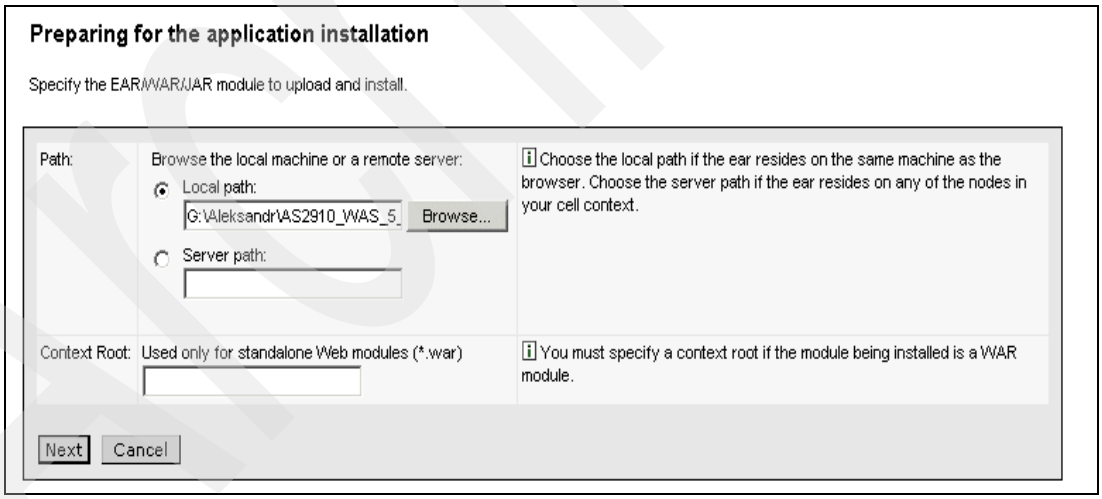


Figure 6-113 Define Path where the EAR file resides

6. On the next page, there are a number of settings which can be changed to customize your application specifically for running on WebSphere. The only settings you will need to change are for the Connection Factory Bindings.

   Select the **Default connection factory bindings** radio button (you may have to scroll down) and enter **eis/jdbc/MyBank_CMP** for the JNDI name. Ensure **Per Connection Factory** is set for the resource authorization. This setting will use the username and password information associated with the DataSource (that is described in step 16 on page 202) when authenticating to the database.

Click **Next**, The application installation will initialize; see Figure 6-114.

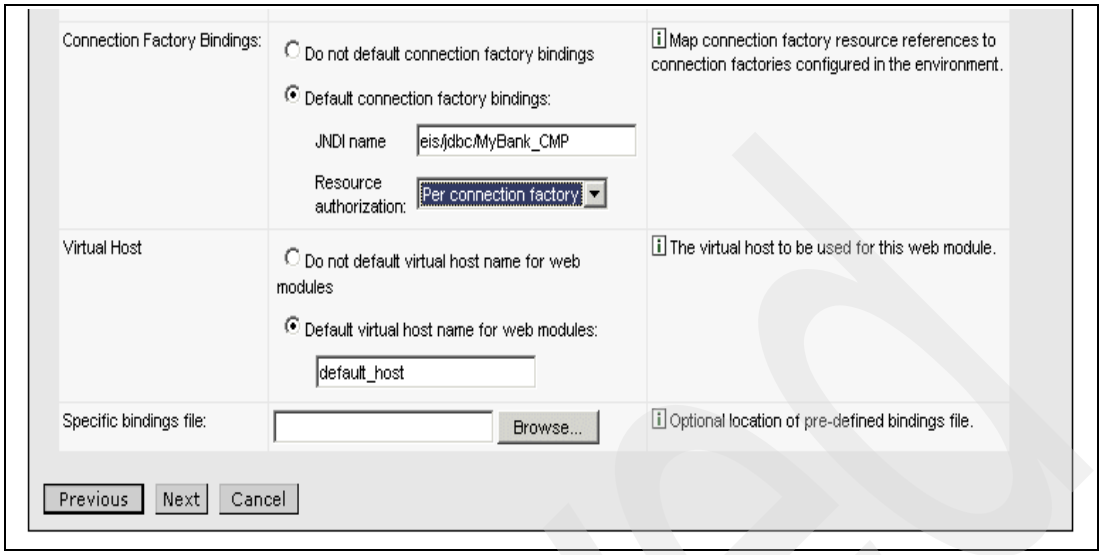


Figure 6-114 Set Connection Factory bindings

Note: The install New Application wizards provides a list of steps; see Figure 6-115. The steps that require user input are marked with a red sign. You can navigate directly to these steps via the link. For all steps that are not marked, no input is required.

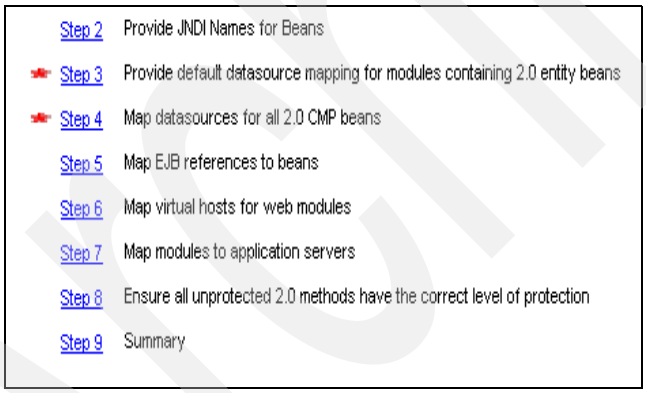


Figure 6-115 Install New Application wizard - list of steps

7. On the Step 1 panel, set the **Deploy EJBs** enable box and make sure that the application name is **MyBankCMP.** Click **Next**; see Figure 6-116.

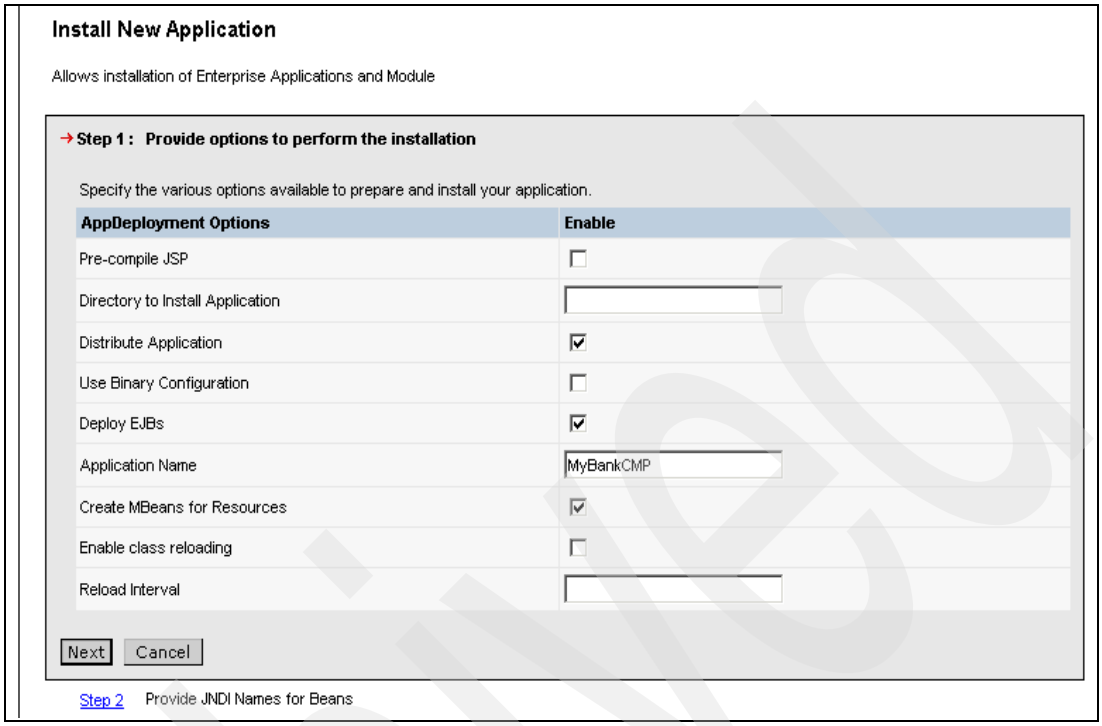


Figure 6-116 Step 1 panel: select Deploy EJB

8. The Step 2 panel has options to use when deploying enterprise beans. We'll accept the default here. Notice that the Database is DB2UDBISeries. Click **Next**.

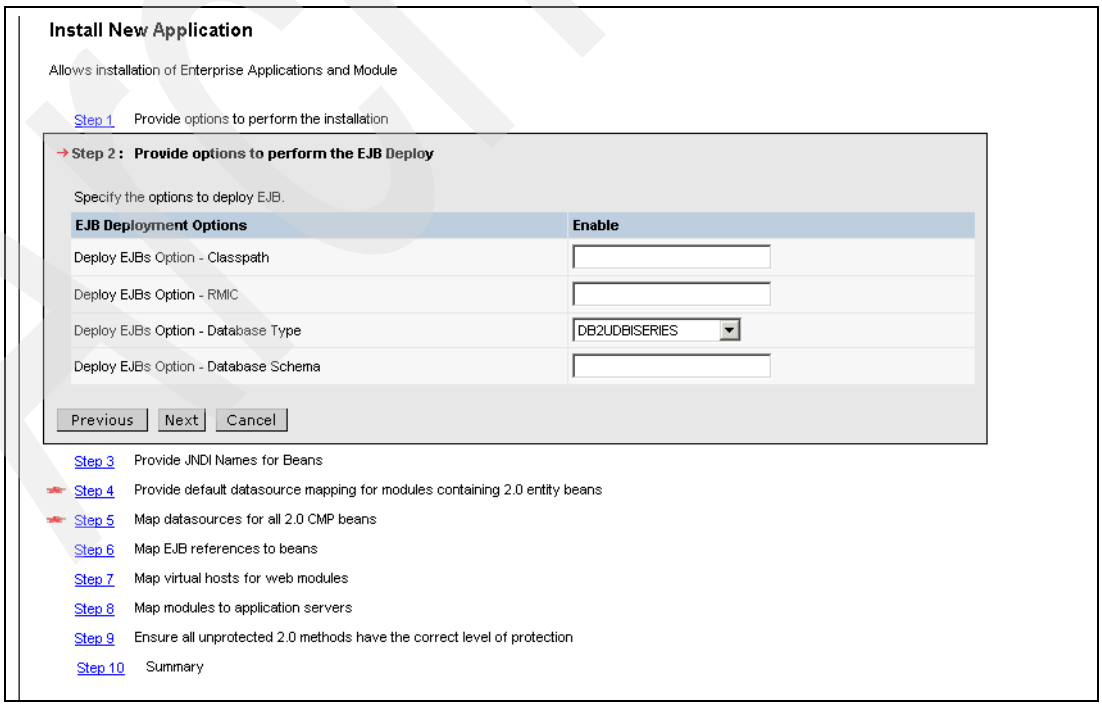


Figure 6-117 Step 2 panel - Database Type

9. In the Step 3 panel, you can specify the JNDI names for the Account and Transfer enterprise beans. Ensure Account and Transfer have been set to **ejb/MyBank/Account** and **ejb/MyBank/Transfer** respectively. Click **Next**; see Figure 6-118.

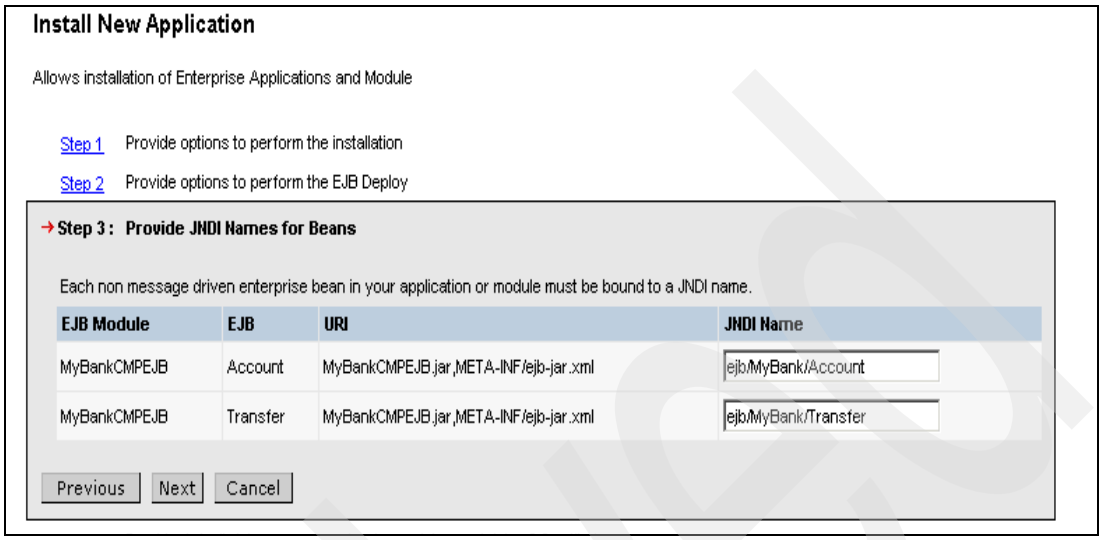


Figure 6-118 Step 3 panel

10. In the Step 4 panel, you can specify a DataSource specifically for your EJB jar file. Select **EJB Module MyBankCMPEJB** and enter **eis/jdbc/MyBank_CMP** for the JNDI Name.

    Select **Per connection factory** from the Resource authorization drop-down list. Click **Next**; see Figure 6-119.

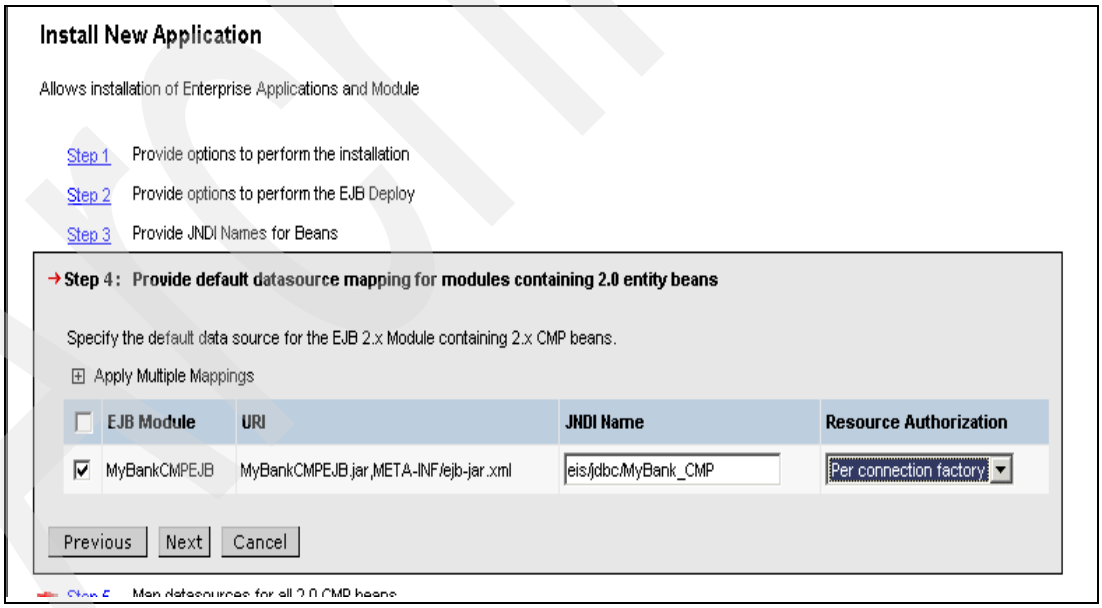


Figure 6-119 Step 4 panel

11. In the Step 5 panel, we can define a DataSource and a resource authorization for each entity EJB. We only have one Entity EJB, but will assign it a DataSource to demonstrate the user interface.

    Expand **Apply Multiple Mappings** (see Figure 6-120).

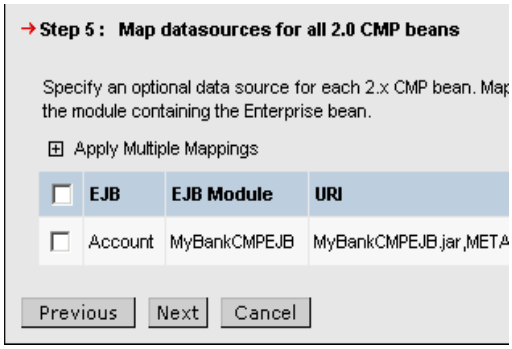


Figure 6-120 Step 5 panel - Apply multiple mappings

a. From the Specify existing Resource JNDI name drop down list, select `nodename_appServerName:eis/jdbc/MyBank_CMP`, where `nodename` is the name of your iSeries server, and `appServerName` is the name of your application server — in our case RCHAS07_bankap:**eis/jdbc/MyBank_CMP**. This is the connection factory that corresponds to the DataSource we have created in step 10 on page 238.

b. Select the checkbox for the EJB Module you want to assign the JNDI name from the list (in our sample, only one EJB is there).

c. Click the associated **Apply** (for the JNDI name). Verify that **eis/jdbc/MyBank_CMP** appears under JNDI Name to the MyBankCMPEJB.

a. On the same panel select **Per Connection Factory** from the **Resource authorization** drop down list and click the checkbox for the EJB Module again.

b. Click the associated **Apply**. Verify that the resource authorization for the Account EJB is **Per Connection Factory**.

c. Click **Next**; see Figure 6-121.

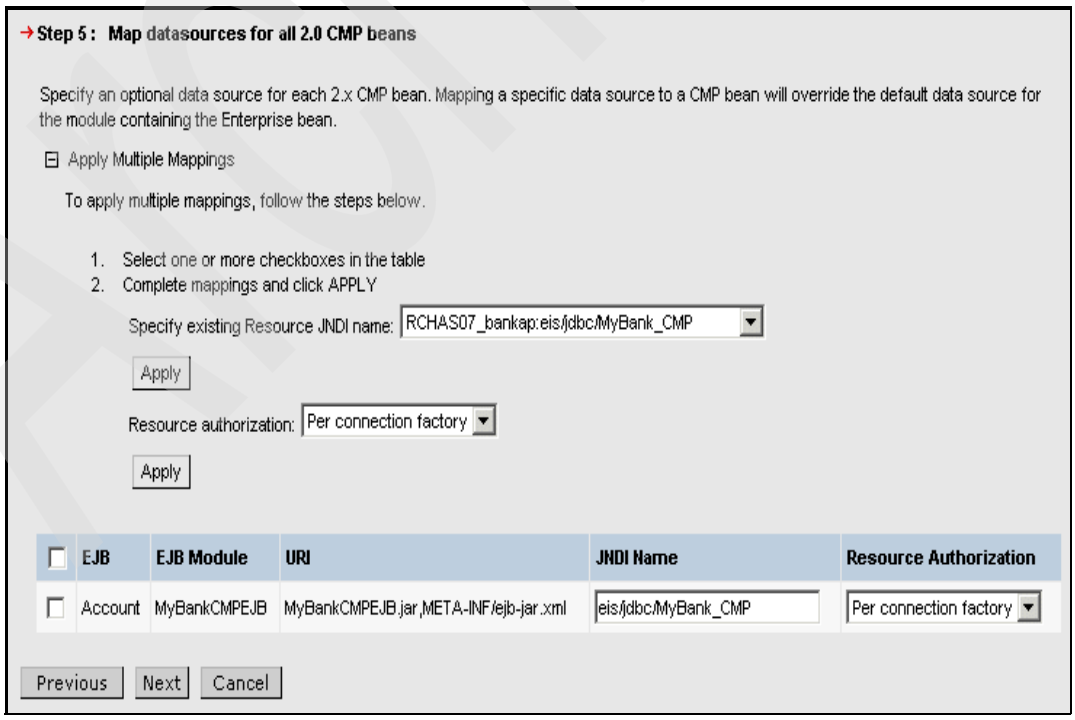


Figure 6-121 Step 5 panel

12. In the Step 6 panel, you can specify EJB References which exist in your application. These references will need to match the JNDI names set for the referenced enterprise beans. Ensure that the JNDI names match the Figure 6-122 and click **Next**.

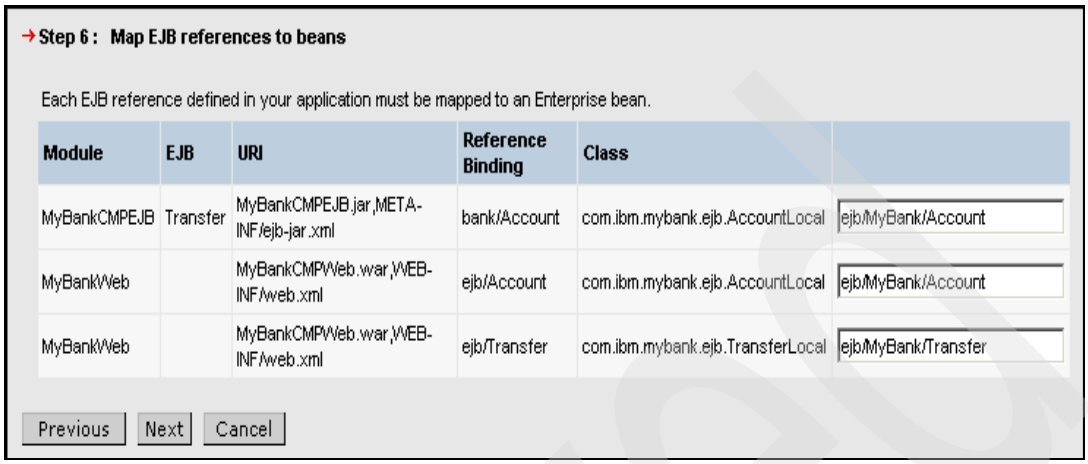


Figure 6-122 Step 6 panel

13. In the Step 7 panel, you can specify the virtual host which the Web module will be assigned to. Check the MyBankWeb and select the virtual host your application will use. Our application works with the default_host. Click **Next**; see Figure 6-123.

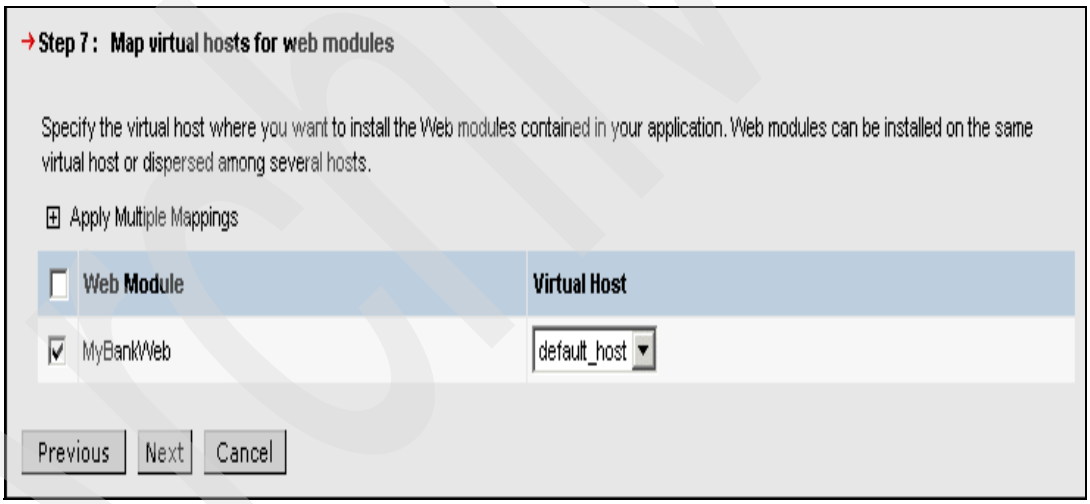


Figure 6-123 Step 7 panel

14. In the Step 8 panel, you can specify which server you would like your application to be installed. Since we are working with a WAS instance with a single application server, we do not have an option of a different server. Accept the default value and click **Next**; see Figure 6-124.

> **Note:** If you have a multiple server environment, you can select the application and assign the appropriate server from the list.

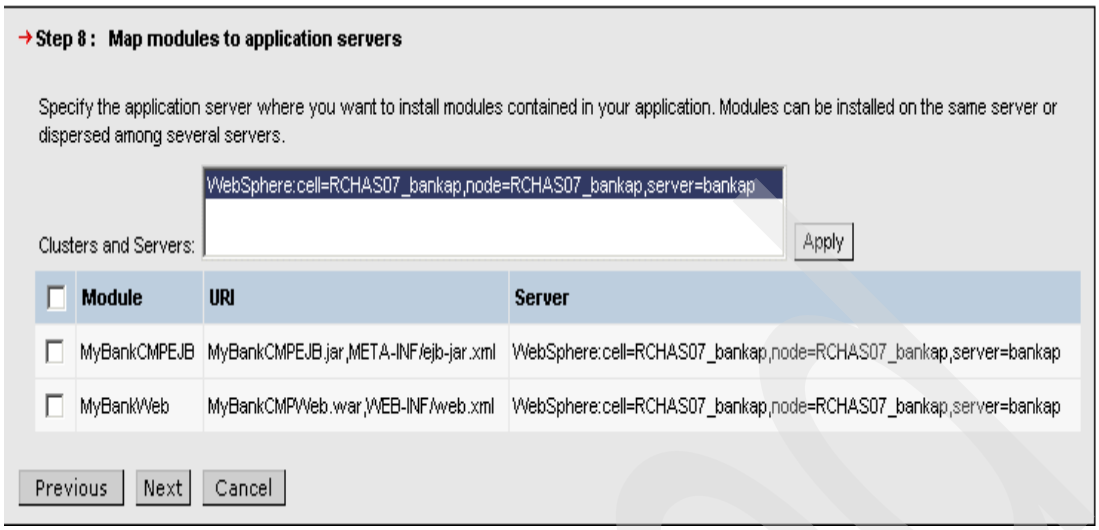


Figure 6-124 Step 8 panel

15. In the Step 9 panel, security can be defined and set for specific methods within your EJB module. We have not set any security roles for the EJB module. Click **Next**; see Figure 6-125.

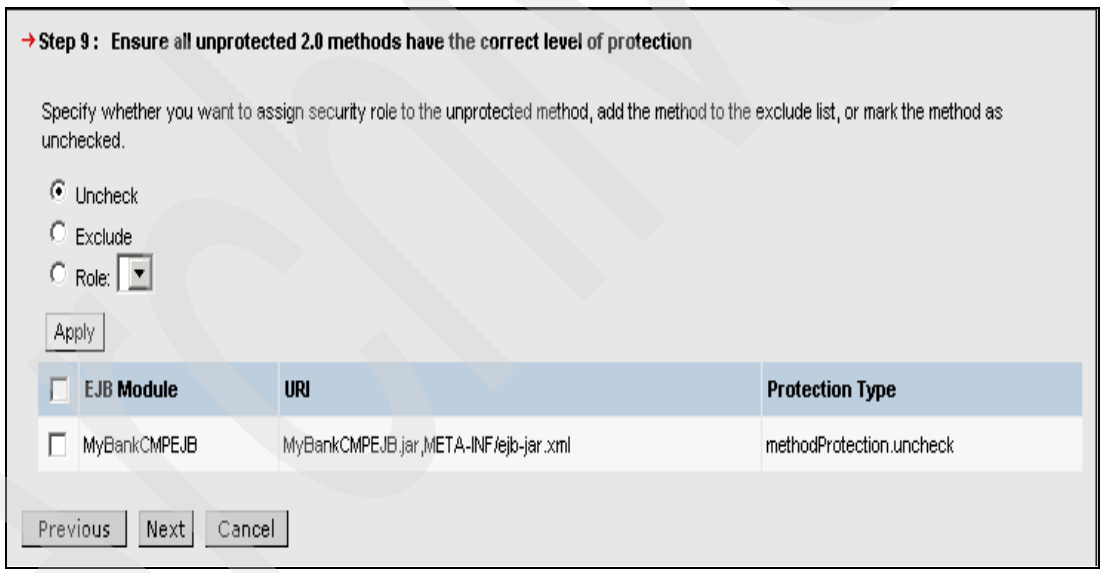


Figure 6-125 Step 9 panel

16. On the Step 10 panel, click **Finish**; see Figure 6-126. The installation will begin. The progress will be displayed in the Workspace window of the console. When the installation has completed, the following message will be displayed:

```
Application MyBankCMP installed successfully.
```

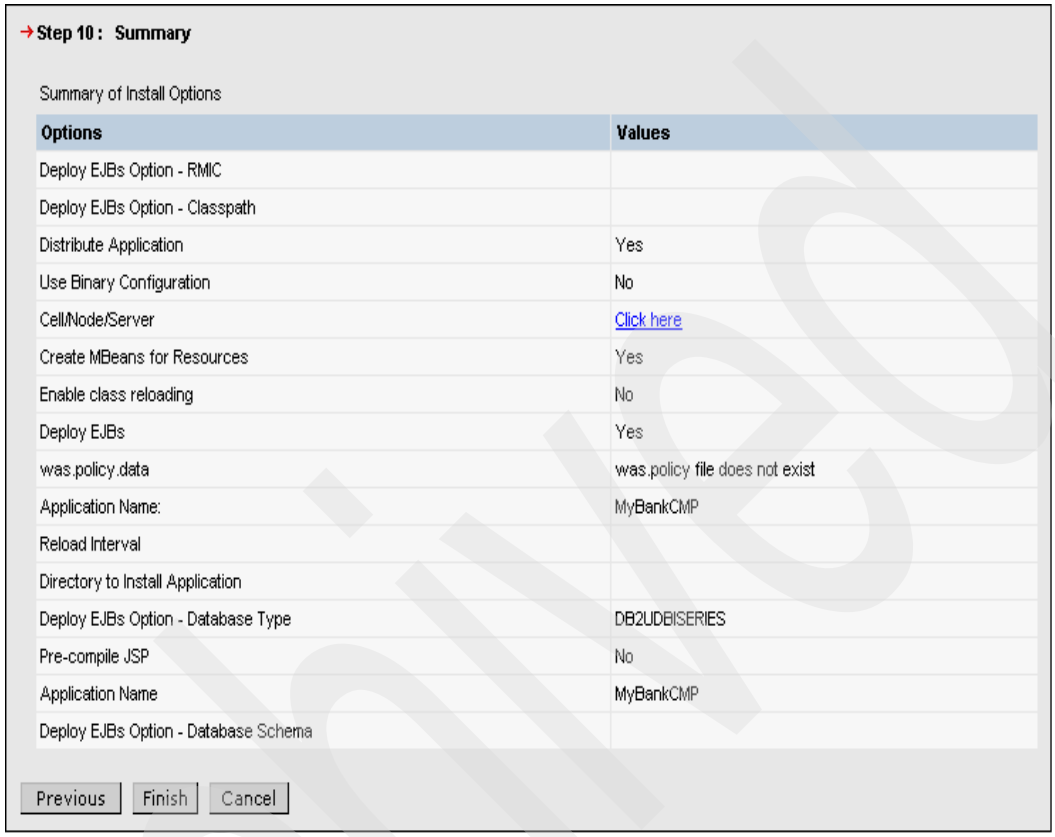


Figure 6-126 Step 10 panel

17. Figure 6-127 shows the sample messages that you will see in the administrative console. You can also find information in the SystemOut.log regarding the instance you want to install the application.

```
Installing..
If there are EJB's in the application, the EJB Deploy process may take several minutes.
Please do not save the configuration until the process is complete.

Check the SystemOut.log on the Deployment Manager or Server where the application is
deployed for specific information about the EJB Deploy process as it occurs.
Starting workbench.
Creating the project.
Building: /MyBankCMPEJB.
Deploying jar MyBankCMPEJB
Creating Top Down Map
Generating deployment code
Building: /MyBankCMPEJB.
Invoking RMIC.
Generating DDL
Generating DDL
Writing output file
Shutting down workbench.
EJBDeploy complete.
0 Errors, 0 Warnings, 0 Informational Messages
ADMA5007I: EJBDeploy completed on /tmp/app_f2094bc766/dpl/dpl_MyBankCMP.ear
ADMA5005I: Application MyBankCMP configured in WebSphere repository
ADMA5001I: Application binaries saved in
/QIBM/UserData/WebAS5/Base/bankap/wstemp/uschi1/workspace/cells/RCHAS07_bankap/applicati
ons/MyBankCMP.ear/MyBankCMP.ear
ADMA5011I: Cleanup of temp dir for app MyBankCMP done.
ADMA5013I: Application MyBankCMP installed successfully.
Application MyBankCMP installed successfully.
If you want to start the application, you must first save changes to the master
configuration.
Save to Master Configuration
If you want to work with installed applications, then click Manage Applications.
Manage Applications
```

Figure 6-127 Installation messages in administrative console

18. When the application has finished installing, save the configuration changes by clicking **Save to Master Configuration**. When the Save window appears, expand **View items with changes**. Notice the different items that will be updated (see Figure 6-128).

19. Click **Save**.

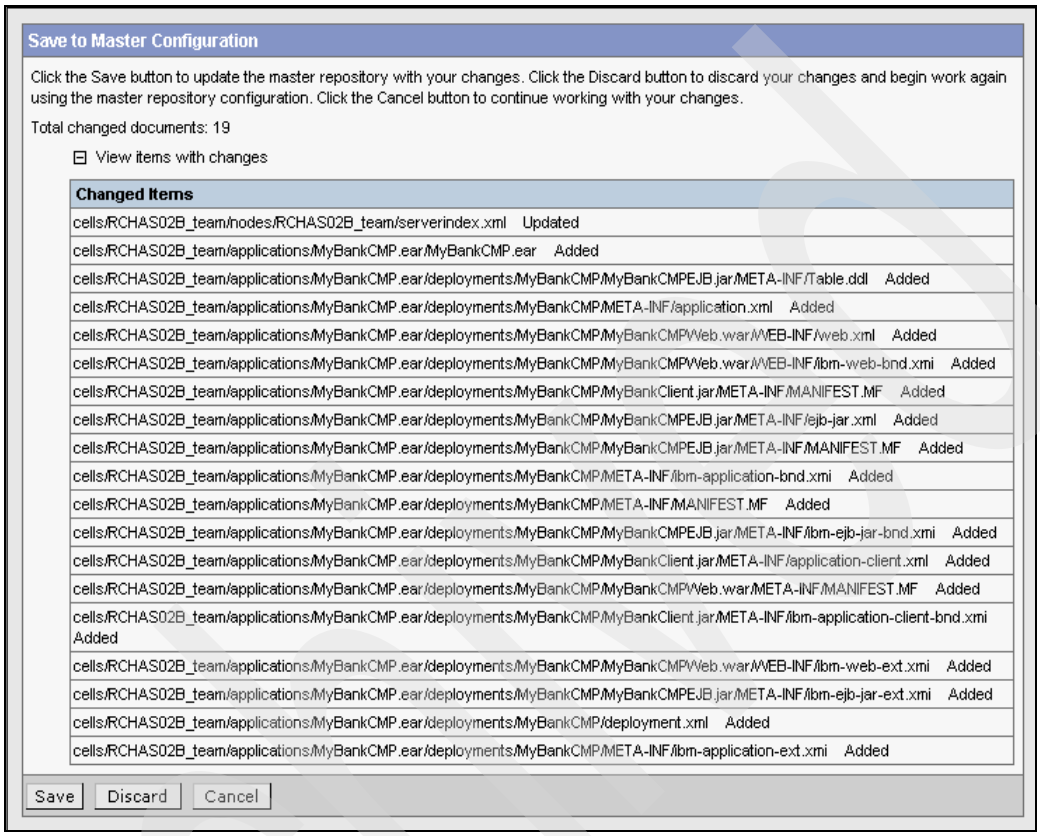


Figure 6-128 View items with changes

20. Because you installed a new application, do the **Update web server plugin**; see 6.7.4, “Updating Web server plug-in configuration” on page 171. After the update, the plugin-cfg.xml looks similar to Figure 6-129. Notice the new entry:

```
<Uri AffinityCookie="JSESSIONID" Name="/MyBankCMPWeb/*"/>.
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Config>
    <Log LogLevel="Error"
Name="/QIBM/UserData/WebAS5/Base/bankap/logs/http_plugin.log"/>
    <VirtualHostGroup Name="default_host">
        <VirtualHost Name="*:10201"/>
        <VirtualHost Name="*:10200"/>
    </VirtualHostGroup>
    <ServerCluster Name="bankap_RCHAS07_bankap_Cluster">
        <Server Name="bankap">
            <Transport Hostname="RCHAS07" Port="10201" Protocol="http"/>
        </Server>
        <PrimaryServers>
            <Server Name="bankap"/>
        </PrimaryServers>
    </ServerCluster>
    <UriGroup Name="default_host_bankap_RCHAS07_bankap_Cluster_URIs">
        <Uri AffinityCookie="JSESSIONID" Name="/snoop/*"/>
        <Uri AffinityCookie="JSESSIONID" Name="/hello"/>
        <Uri AffinityCookie="JSESSIONID" Name="/hitcount"/>
        <Uri AffinityCookie="JSESSIONID" Name="*.jsp"/>
        <Uri AffinityCookie="JSESSIONID" Name="*.jsv"/>
        <Uri AffinityCookie="JSESSIONID" Name="*.jsw"/>
        <Uri AffinityCookie="JSESSIONID" Name="/j_security_check"/>
        <Uri AffinityCookie="JSESSIONID" Name="/servlet/*"/>
        <Uri AffinityCookie="JSESSIONID" Name="/ivt/*"/>
        <Uri AffinityCookie="JSESSIONID" Name="/MyBankCMPWeb/*"/>
    </UriGroup>
    <Route ServerCluster="bankap_RCHAS07_bankap_Cluster"
        UriGroup="default_host_bankap_RCHAS07_bankap_Cluster_URIs"
VirtualHostGroup="default_host"/>
```

Figure 6-129 plugin-cfg.xml after updating Web server plugin

21. With the MyBankCMP application fully installed, as well as the DataSource defined (see “Configuring JDBC Providers and DataSource” on page 195), you can start the MyBankCMP application via the administrative console, in the same way as described in “Starting your new MDB application” on page 241. The next step is to test the application, as described in the following section.

Testing the MyBankCMP sample application

Perform these steps to test the application:

1. Go to a Web browser and access the MyBankCMP application with the following URL:
   - Using the WAS internal HTTP server:

     ```
     http://server_name:internal_HTTPport/MyBankCMPWeb/index.html
     ```

     In our case:

     ```
     http://rchas07:10201/MyBankCMPWeb/index.html
     ```

   - Using the external HTTP server:

     ```
     http://server_name:external_HTTPport/MyBankCMPWeb/index.html
     ```

     In our case:

     ```
     http://rchas07:10200/MyBankCMPWeb/index.html
     ```

You will get a page like the one shown in Figure 6-130.

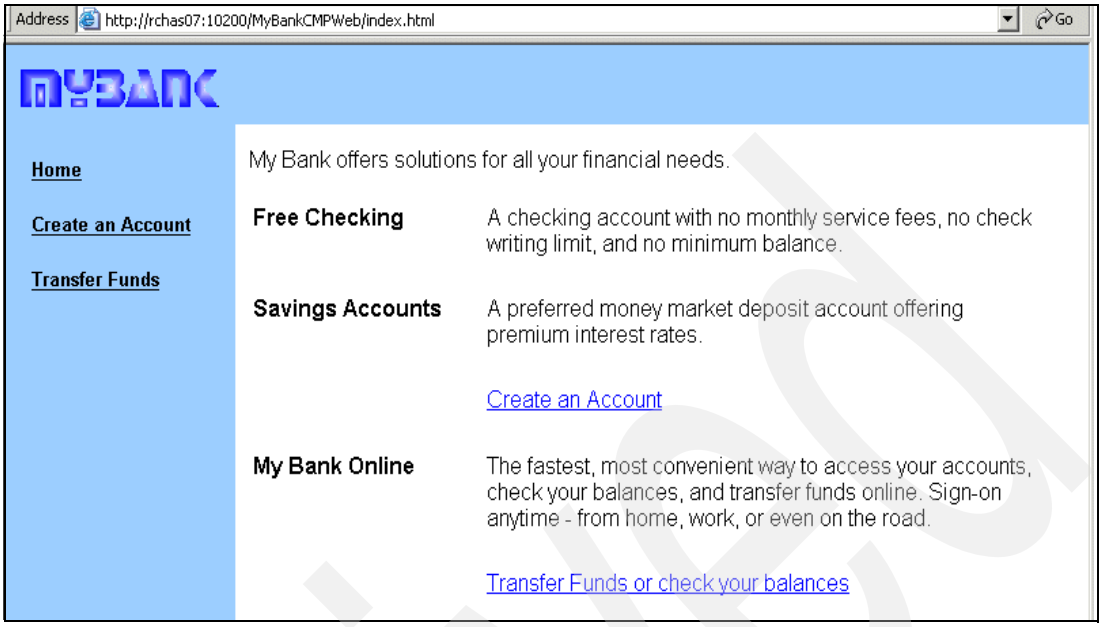


Figure 6-130 MyBankCMP Start page

2. Click **Create an Account** link.

   Enter an Account Number, for example **10,** and for the Starting Balance, enter **1000**, click **Create**; see Figure 6-131.

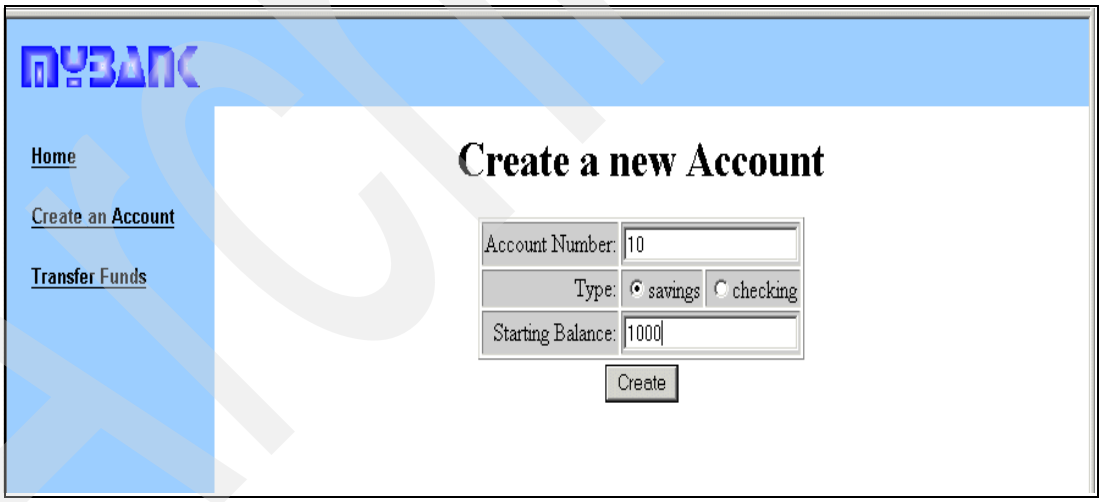


Figure 6-131 Create a new account

3. You will see that the account is created; see Figure 6-132.

Figure 6-132 Bank account created

4. Create the second account, use a Account Number of **20** and a Starting Balance of **4000** and click **Create**.
5. Click the **Transfer Funds** link.
6. Enter **500** for the Amount and **20** for the **From** Account and **10** for the **To** Account. Click **Transfer**; see Figure 6-133.

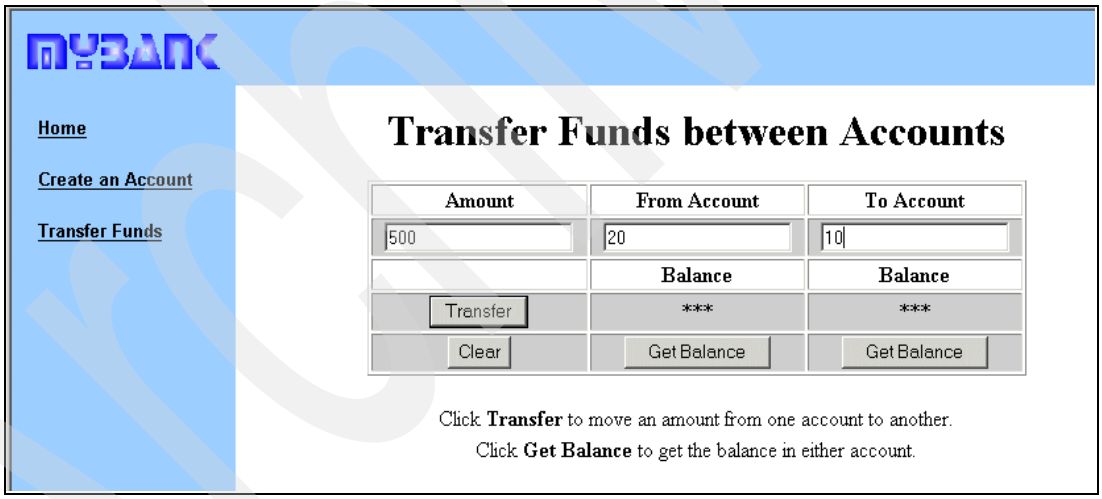


Figure 6-133 Transfer funds

7. You will get a page similar to the one shown in Figure 6-134.

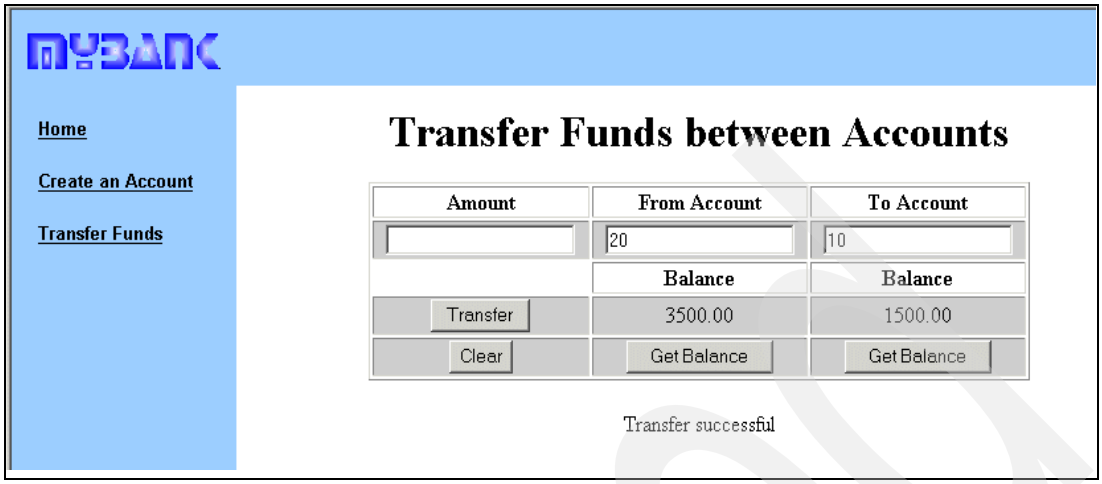


Figure 6-134 Transfer successful

6.13.3 JMS Administration in WebSphere 5.0

Java Message Service (JMS) supports the development of message-based applications in the Java programming language, allowing for the asynchronous exchange of data and events throughout an enterprise.

JMS provides a common mechanism for Java programs (clients and J2EE applications) to create, send, receive, and read asynchronous requests, as JMS messages. This support enables J2EE applications, as JMS clients, to exchange messages asynchronously with other JMS clients by using JMS destinations (queues or topics). A J2EE application can explicitly poll for messages on a destination, then retrieve messages for processing by business logic beans.

Figure 6-135 shows an application polling a JMS destination to retrieve an incoming message, which it processes with a business logic bean. The business logic bean uses standard JMS calls to process the message.

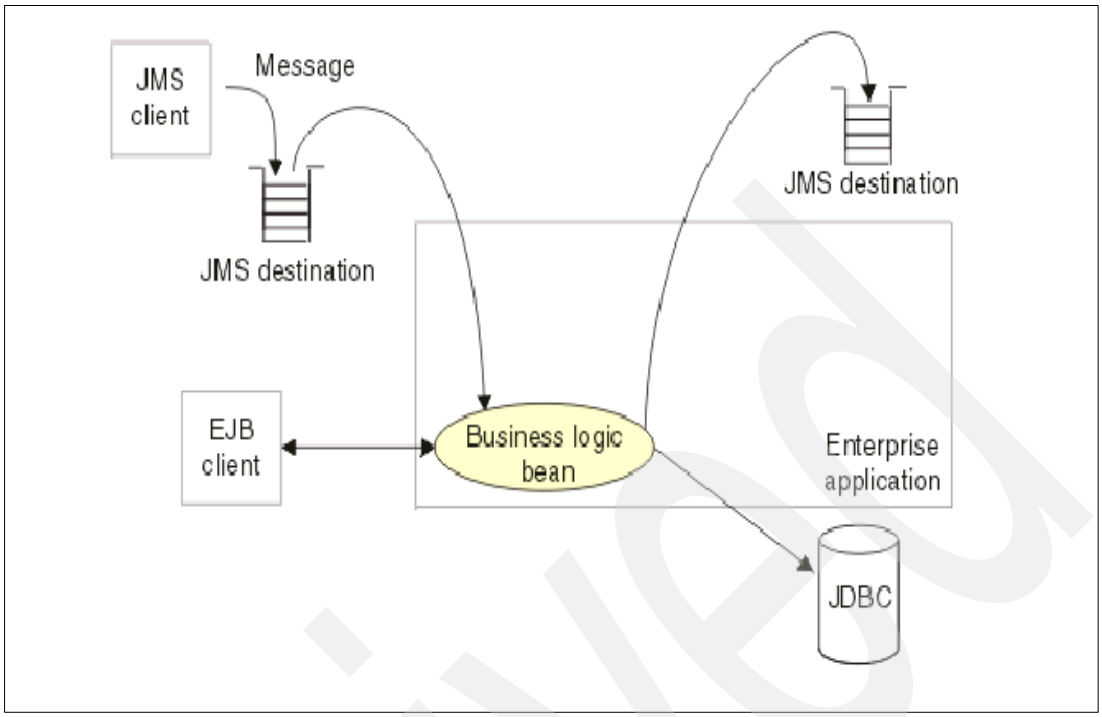


Figure 6-135 Asynchronous messaging using JMS

For information on which components you have to install on your iSeries server to support JMS, refer to 2.5, “Considering if you need Java Messaging Service (JMS)” on page 18.

JMS components

The components of the Java Message Service are as follows:

- The JMS provider is a base messaging system and related Java classes that implement the JMS API.
- The JMS server within the application server accesses the JMS functions of the JMS provider.

  WebSphere Application Server supports three types of JMS resources:

  – WebSphere JMS Server Resources are related to the built in JMS server (embedded JMS server)

  – WebSphere MQSeries Resources are related to the full WebSphere MQ Series 5.3 product.

  – Generic JMS Server Resources are related to third party JMS servers that you may want to configure.

  The embedded JMS server itself is defined differently for Base and Network Deployment environment. In the base configuration, JMS server runs within the Application Server's JVM. In the Network Deployment edition, the JMS server is started in its own JVM. Only one JMS server can be defined on a given node.

  A JMS server is responsible for managing:

  - Queue manager, which is the provider of the point-to-point (queue) service
  - Broker, which is the provider of the publish/subscribe (topic) service

- JMS administered objects are created by the administrator of a WAS environment; see “JMS administered objects” on page 222.
  - The Java Naming and Directory Interface (JNDI) namespace is used to hold references to JMS administered objects.

JMS administered objects

JMS resources are the key to JMS portability. JMS resources are created by a JMS administrator and are specific to the underlying JMS-vendor implementation. These JMS resources, however, implement standard JMS interfaces and are retrieved by JMS applications through JNDI. JMS developers must only know the JNDI name of the JMS resources, and do not have to write any vendor-specific code to use the administered objects.

In order to send a JMS message from an application, we need the following JMS resources:

- JMS connection factory: This is used to create connections with the JMS provider for a specific JMS queue or topic destination. There are two types of JMS connection factories:
  - Queued Connection Factory encapsulates the settings necessary to connect to a queue-based messaging system
  - Topic Connection Factory encapsulates the settings necessary to connect to a topic-based messaging system
- JMS destinations objects provide a destination address for a message. They are used by the application to send and receive messages. In JMS, messages are sent to destinations, not to other applications directly.

  There are two types of destination objects:
  - Topic, the publish/subscribe type: In this one-to-many messaging model, message producers called publishers and message consumers called subscribers communicate through virtual channels called topics. One sender submits a message to a topic, and all of the clients that are subscribed to that topic can receive the message. JMS publish/subscribe uses topic connection factory and topic resources to communicate with the JMS provider.
  - Queue, point-to -point type: Point-to-point applications use queues to pass messages between each other. In this one-to-one messaging model, message producers called senders and message consumers called receivers communicate through virtual channels called queues. Each message has one sender and one recipient. A client sends a message to a specific queue and the message is picked up and processed by a server listening on that queue. JMS point-to-point uses queue connection factory and queue resources to communicate with the JMS provider.

Note: Both styles of messaging can be used in the same application.

The WebSphere administrative console can be used to configure and administer these resources for the embedded JMS provider and the WebSphere MQ JMS provider.

Note: The functionality of the administrative console is twofold for the embedded and WebSphere MQ JMS Provider. With the administrative console, you can:

- Configure JMS administrative objects in the WebSphere name space.
- Configure and create messaging resources (queues) in the messaging system.

Support for message driven beans (MDBs)

An application server product that is compliant with the J2EE 1.3 specification, like WebSphere Application Server V5.0 for iSeries, must provide support for automatic asynchronous messaging using message driven beans (MDBs), a type of enterprise bean defined in the EJB 2.0 specification.

The support for MDBs is based on the JMS message listener, which comprises the following components:

- Listener manager, which controls and monitors one or more listeners.
- Listener, a JMS destination for incoming messages:

  When a message arrives on the destination, the listener passes the message to a new instance of a user-developed MDB for processing. The listener then looks for the next message without waiting for the bean to return.

  Depending on the JMS messaging model the listener uses, it is associated with either:

  – A queue connection factory and a queue
  – A topic connection factory and a topic

  You have to configure the message listener service before you install MDB application components.
- JMS destination, which represents a queue or topic.
- Message driven beans (MDBs):

  A stateless component that is invoked by the J2EE container as a result of the arrival of a JMS message at a particular JMS destination (queue or topic). The MDB is, in a sense, triggered by the arrival of the message.

  Messages arriving at a destination being processed by a listener have no client credentials associated with them: the messages are anonymous. To enable some level of security, a listener assumes the credentials of the application server process for the invocation of the MDB.

  MDBs can handle messages read from JMS destinations within the scope of a transaction. If transaction handling is specified for a JMS destination, the JMS listener starts a global transaction before it reads any incoming message from that destination. When the MDB processing has finished, the JMS listener commits or rolls back the transaction (using JTA transaction control).

Figure 6-136 shows an incoming message being passed by a JMS listener to a MDB, which passes the message on to a business logic bean for business processing. This messaging is controlled by the listener manager.

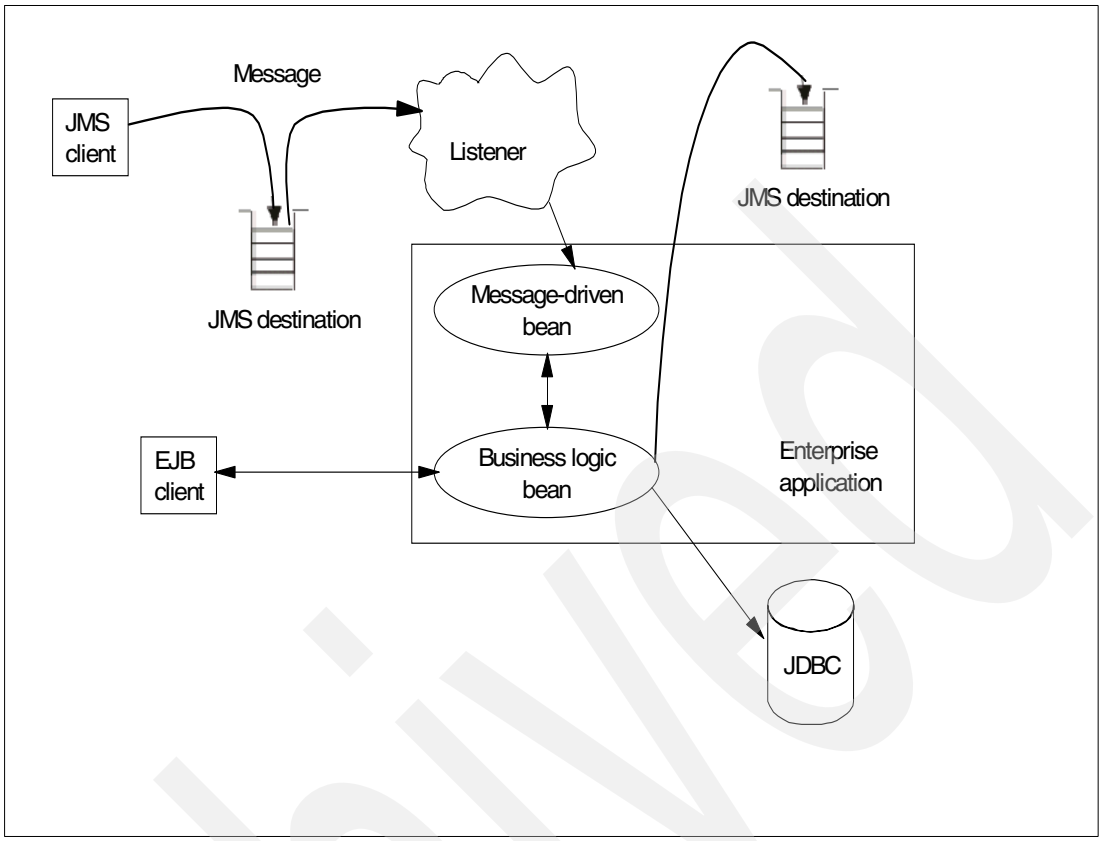


Figure 6-136 Support for MDBs

Enabling your instance to use Java Message Service (JMS)

In this section we describe the steps that are necessary to configure WebSphere Application Server to work with the embedded JMS provider. A JMS provider is used in our message-driven bean (MDB) sample application named `jmsmdb`.

We added some general information in the description of these steps, so you can follow them also to implement the necessary resources for your own application.

The message-driven beans sample application is a part of the WebSphere Application Server samples gallery, which provides a set of small, generic samples that show how to perform common enterprise application tasks. You can also access the online documentation for more information at:

`http://publib.boulder.ibm.com/iseries/v1r1m0/websphere/ic2924/index.htm?info/rzaiz/50/was.htm`

These are the basic steps to enable our instance to use embedded JMS:

- Setting up new WAS instance and user profile for the sample MDB application, starting at page 225.

  You can also use the WAS default instance and configure the embedded JMS support for this instance.
- Creating a new J2C Authentication Data Entry, starting at page 226.
- Configuring your JMS connection factory, starting at page 228.
- Configuring your JMS topic destinations, starting at page 231.

- Configuring your JMS listener port, starting at page 233.
- Deploying your MDB application, starting at page 236.
- Starting your new MDB application, starting at page 241.

Setting up new WAS instance and user profile for the sample MDB application

The first part of the example is to create a new WAS instance:

1. Create a new WAS instance with the `crtwasinst` script (see Figure 6-137), in which our message-driven beans sample will be deployed and run. We named this instance jmsmdb. For more information how to create an instance, see “Creating a new WAS instance” on page 113.

```
crtwasinst -instance jmsmdb -exthttp 10400 -inthttp 10401 -admin 10402 -portblock 10405
Creating instance jmsmdb...
 ADCP0005I: Using cell RCHAS07_jmsmdb, node RCHAS07_jmsmdb and server jmsmdb.
 ADCP0006I: Embedded JMS enabled.
 Instance jmsmdb created.
 Instance root directory is /QIBM/UserData/WebAS5/Base/jmsmdb.
 The application server name is jmsmdb.
 Ports:
   External HTTP:  10400
   External HTTPS: 443
   Name service:   10405
   JMS secure:     10406
   JMS queued:     10407
   JMS direct:     10408
   DRS client:     10409
   SOAP:           10410
   SAS:            10411
   CSIV2 Mutual:   10412
   CSIV2 Server:   10413
   Internal HTTP:  10401
   Admin:          10402
   Admin SSL:      10414
   CSIV2 Server:   10413
   Internal HTTP:  10401
   Admin:          10402
   Admin SSL:      10414
  $
```

Figure 6-137 Create jmsmdb instance

2. Start the WAS instance jmsmdb with the `startServer` script. For more information, see 6.5.3, “Starting a specific application server” on page 117.

   Figure 6-138 shows how we started our jmsmdb instance.

```
startServer jmsmdb -instance jmsmdb
CPC1221: Job 037934/QEJBSVR/JMSMDB submitted to job queue QEJBJOBQ in
library QEJBAS5.
EJB6123: Application server started.
 Cause . . . . . :   Application server jmsmdb in Base instance jmsmdb has
   started and is ready to accept connections on admin port 10402.
$
```

Figure 6-138 start jmsmdb instance

3. An HTTP server for running the MDBSamples application is not necessary, because the JMS client communicates with the MDB through the listener manager.
4. Create an iSeries user profile JMSUSER with password JMSWORK.

   This user profile is used for JMS connection factories.
5. All resources you need to enable your instance to use JMS are done by the administrative console. Start your administrative console for your instance; see 6.7, “Working with the Administrative Console” on page 159.

   Because our jmsmdb instance has the admin port 10402 defined; see Figure 6-137 on page 225, we use:

   ```
   http://rchas07:10402/admin
   ```

Creating a new J2C Authentication Data Entry

The next step in the example is to configure the authentication information:

1. Create a new J2C Authentication Data Entry for use with JMS connection factories (Java 2 Connector security).

   In the topology tree, expand **Security -> JAAS Configuration**, and click **J2C Authentication Data**; see Figure 6-139.

Figure 6-139 Create J2C Authentication Data Entry

2. The J2C Authentication Data Entries panel lists the existing aliases. Click **New**; see Figure 6-140.

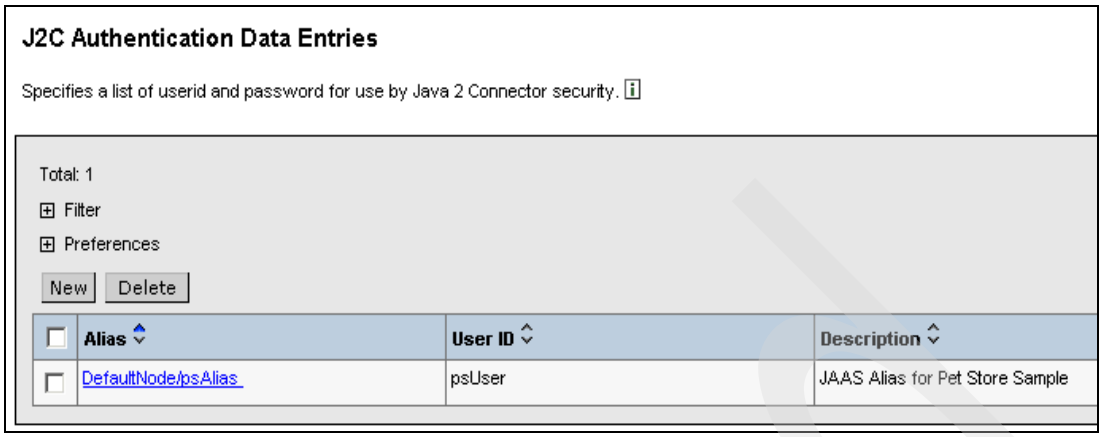


Figure 6-140 J2C Authentication Data Entries

3. In the Alias field, specify the name of the authentication data entry.

   You can use any name for the Alias.

   For our message-driven beans sample application, we specify **mdb**.

4. Specify any valid user profile and password.

   We use **JMSUSER** for the userid and **JMSWORK** for the password.

5. We use **J2C Authent for MDB sample** for the description.

> **Note:** The user profile and password you specify must also be a valid iSeries server user ID and password.

6. Click **OK**; see Figure 6-141.

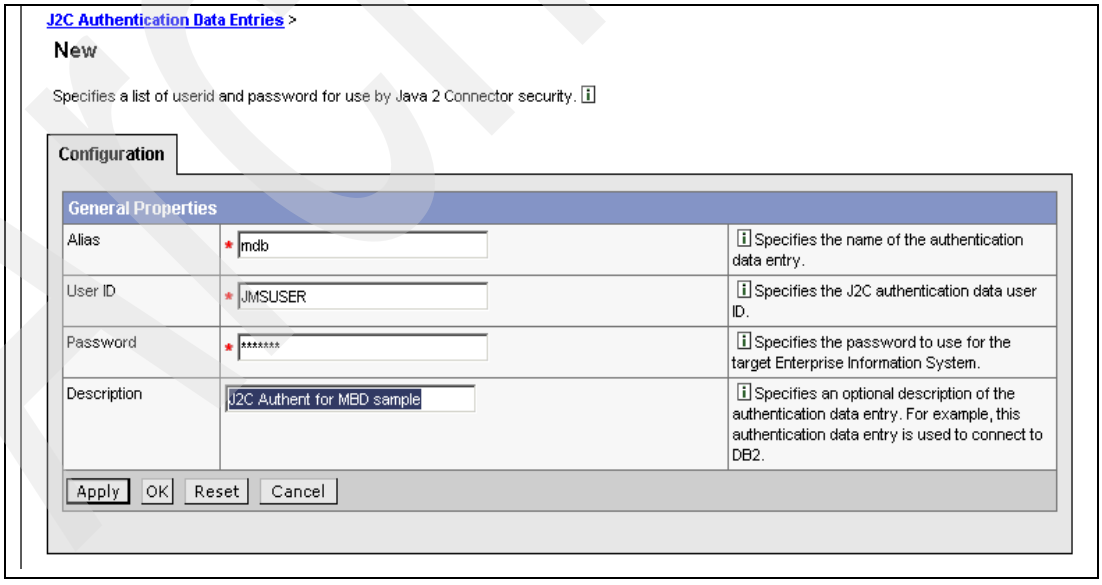


Figure 6-141 J2C Authentication Data Entries - General Properties

7. You will get a panel like the one shown in Figure 6-142.
8. In the Messages box at the top of the page, click **Save**; see Figure 6-142.

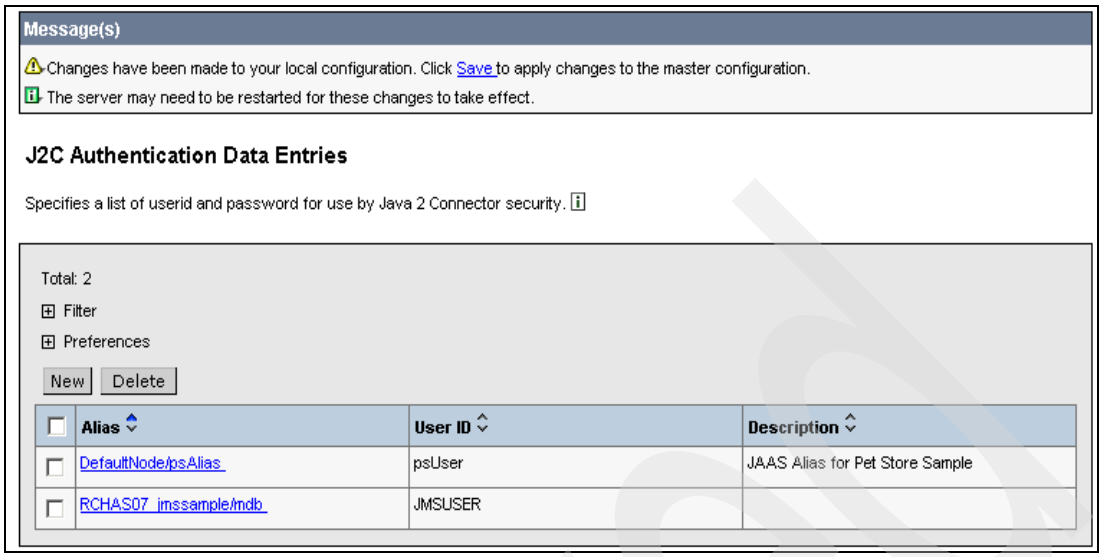


Figure 6-142 J2C Authentication Data Entries - new

9. Click **Save** to save your changes to the master configuration.

Configuring your JMS connection factory

In our example, we work with the WebSphere embedded JMS server.

WebSphere MQ JMS resources can also be administered and configured via the administrative console and is similar to the support for the embedded JMS server. You define the WebSphere MQ JMS resources under the **WebSphere MQ JMS Provider** link:

1. In the topology tree, expand **Resources**, and click **WebSphere JMS Provider**; see Figure 6-143.

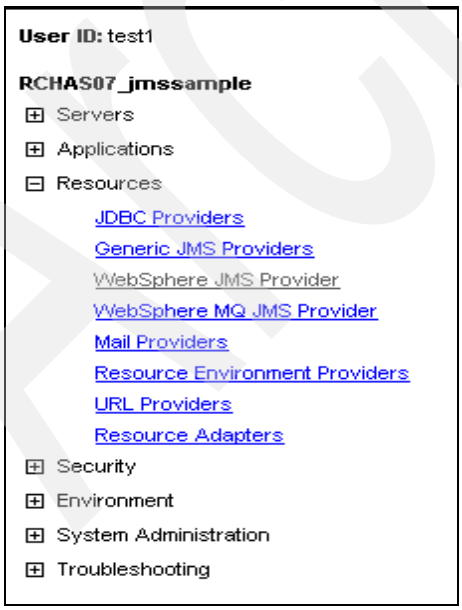


Figure 6-143 WebSphere JMS provider

2. Our example uses a topic connection factory to create connections to the associated JMS provider of JMS topic destinations for publish and subscribe messaging.

   For a JMS queue destination, you would use a queue connection factory.

   Scroll down to the Additional Properties and click **WebSphere Topic Connection Factories**; see Figure 6-144.

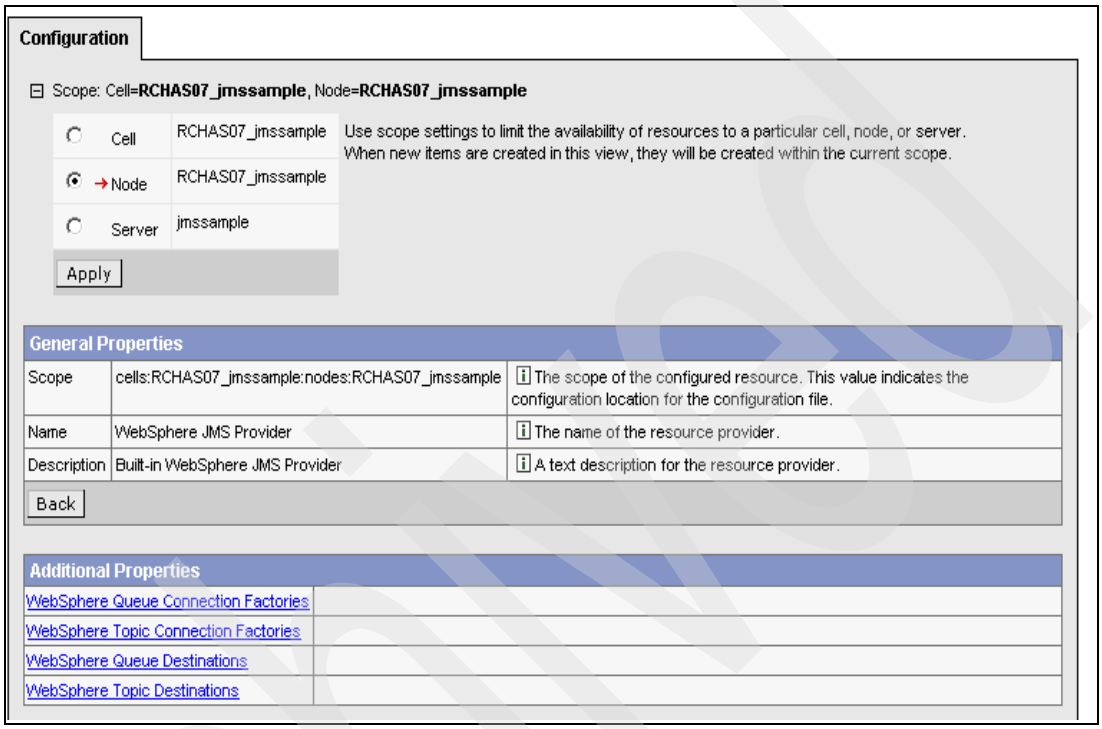


Figure 6-144 JMS Provider WebSphere Topic Destinations

3. Click **New**; see Figure 6-145.

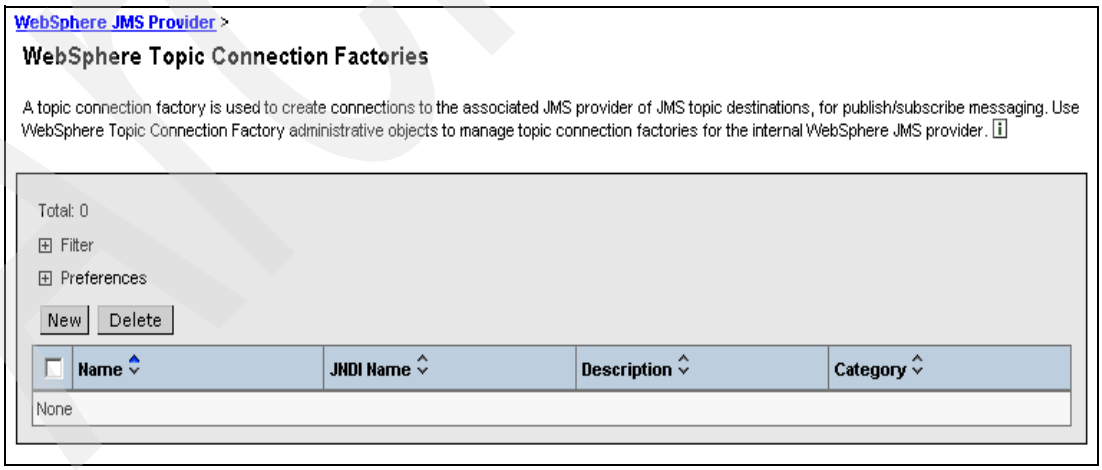


Figure 6-145 WebSphere Topic Connection Factories

4. In the Name field, specify **SampleJMSTopicConnectionFactory**.

   In the JNDI name field, specify **Sample/JMS/TCF**. This name is later used when you define the JMS listener port.

   For the Description, we use **MDB sample TopicConnectionFactory**.

For both the **Component-managed Authentication Alias field** and the **Container-managed Authentication Alias field** select ***nodeName*_jmsmdb/mdb** from the corresponding drop-down box.

This is the Authentication Data Entry you created in the previous step see “Creating a new J2C Authentication Data Entry” on page 226.

In the Client ID field, specify **MDBSampleClientID**; see Figure 6-146 and Figure 6-147.

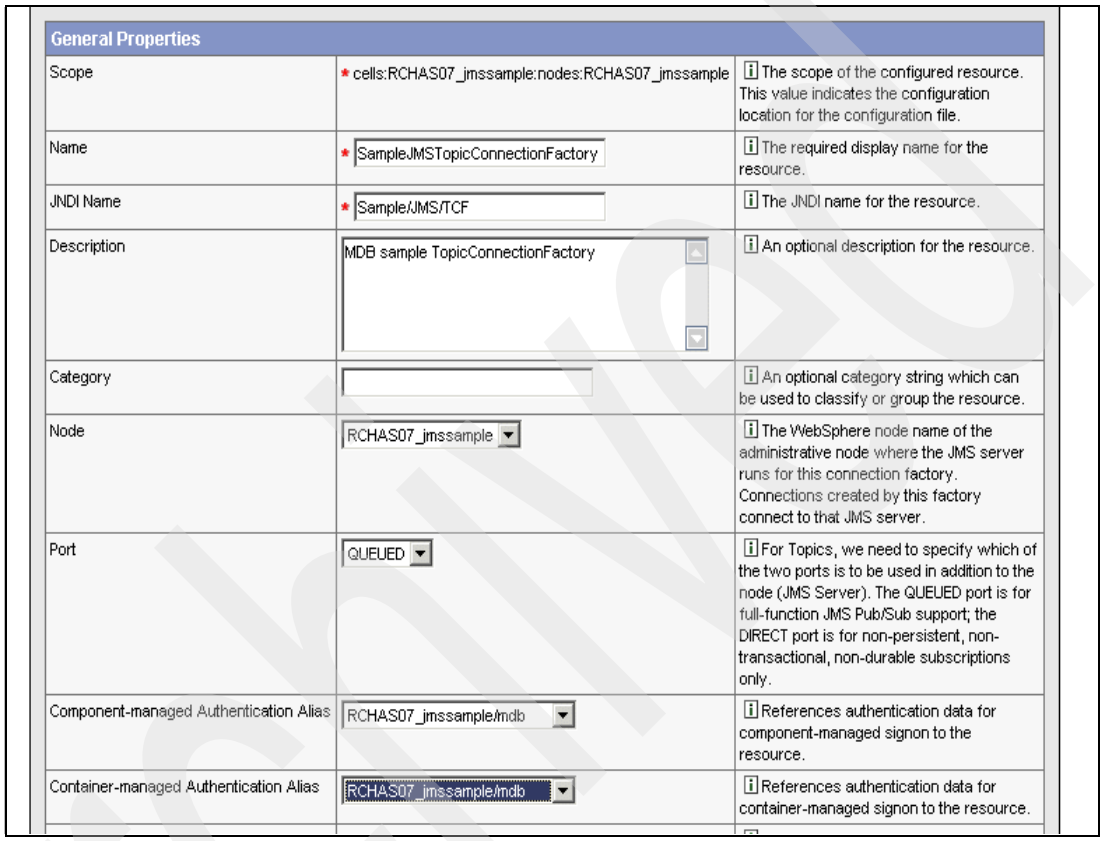


Figure 6-146 WebSphere Topic Connection Factories definition part 1

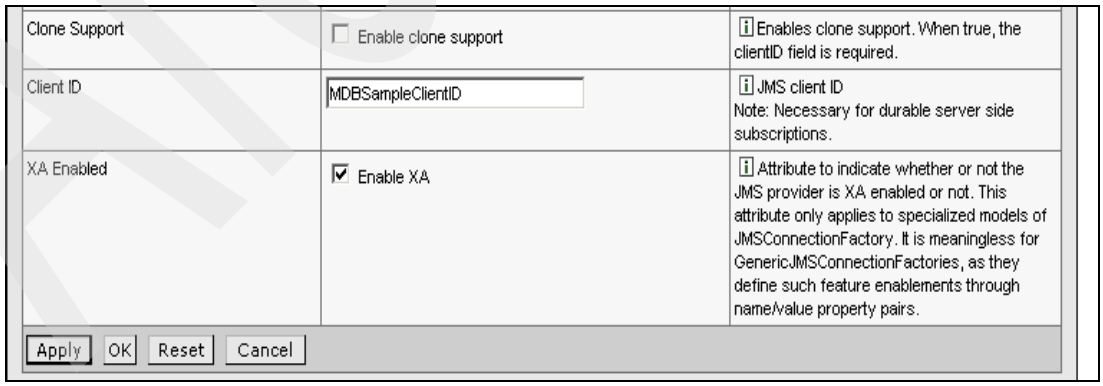


Figure 6-147 WebSphere Topic Connection Factories definition part 2

5. Click **OK**; see Figure 6-147.
6. You should see your connection factory created (see Figure 6-148).

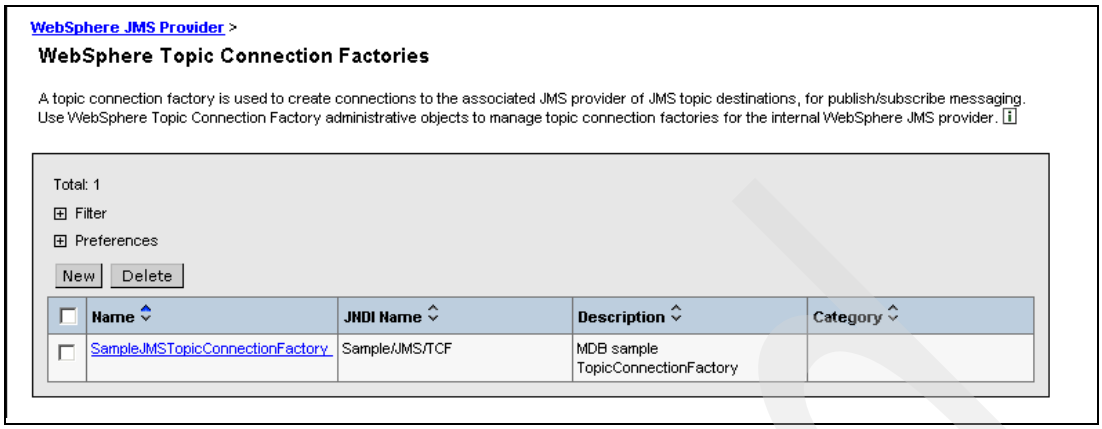


Figure 6-148 New JMS connection factory

Configuring your JMS topic destinations

Topic destinations are provided for publish/subscribe messaging by the internal WebSphere JMS provider. To send messages, applications publish messages to topics. To receive messages, applications subscribe to topics. When a message is published to a topic, it is automatically sent to all the applications that are subscribers to that topic.

Use WebSphere Topic Destination administrative objects to manage topic destinations for the internal WebSphere JMS provider.

For the message-driven beans sample application, we create four separate topics.

1. In the topology tree, expand **Resources**, and click **WebSphere JMS Provider**.
2. Scroll down to the bottom of the page, and click **WebSphere Topic Destinations**.
3. Click **New**; see Figure 6-149.

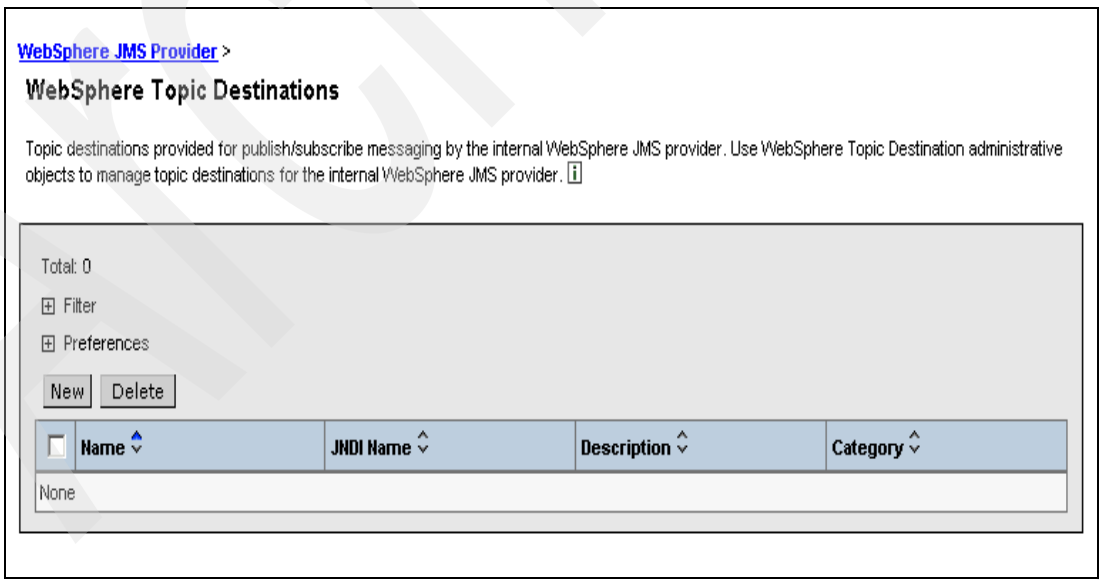


Figure 6-149 WebSphere Topic Destinations - New

4. In the Name field, specify **Sample.JMS.listen**.
   - In the JNDI name field, specify **Sample/JMS/listen**.
   - In the Topic field, specify **Sample/JMS/listen**.

The JNDI name you specify here for the destination is later used when you configure the JMS listener port.

5. Click **OK**; see Figure 6-150.

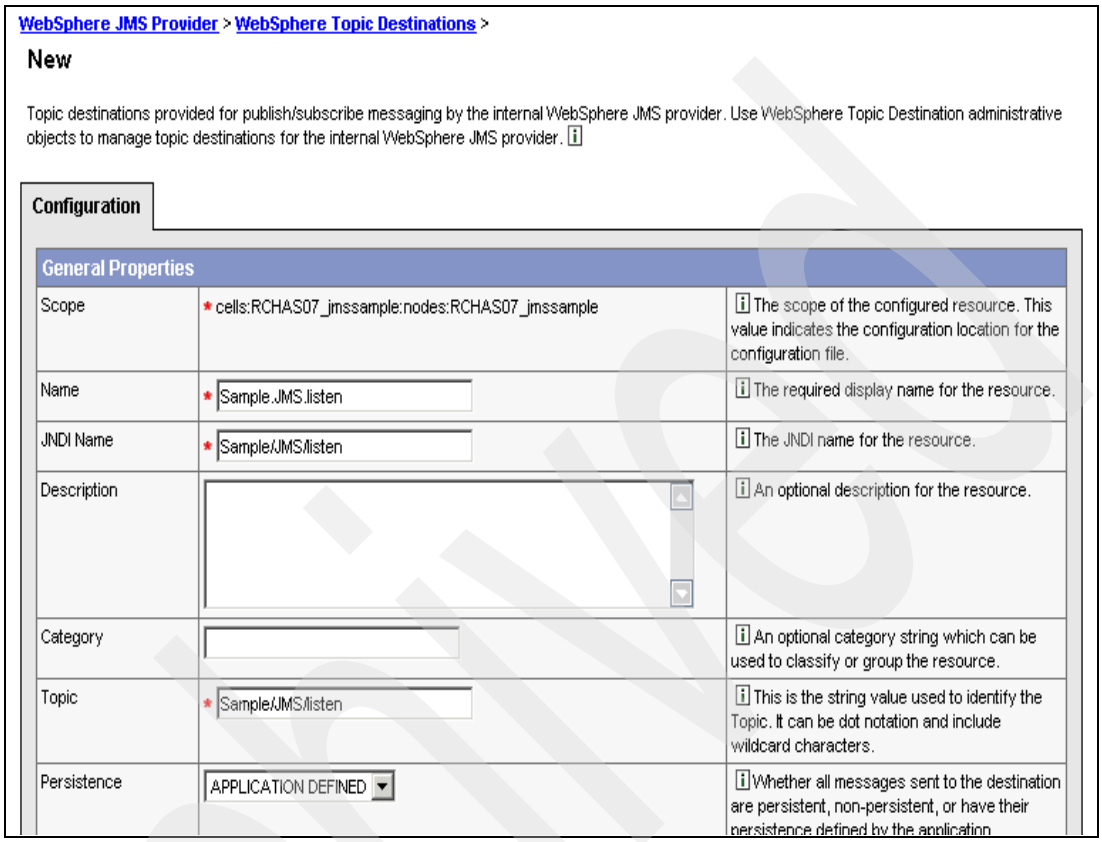


Figure 6-150 WebSphere Topic Destinations

6. Repeat the steps 3 on page 231 and 4 on page 231 to create the next topic, using the following values:
   - In the Name field, specify **Sample.JMS.news**.
   - In the JNDI name field, specify **Sample/JMS/news**.
   - In the Topic field, specify **Sample/JMS/news**.
7. Click **OK**.
8. Repeat the steps 3 on page 231 and 4 on page 231 to create the next topic, using the following values:
   - In the Name field, specify **Sample.JMS.sport**.
   - In the JNDI name field, specify **Sample/JMS/sport**.
   - In the Topic field, specify **Sample/JMS/sport**.
9. Click **OK**.
10. Repeat the steps 3 on page 231 and 4 on page 231 to create the next topic, using the following values:
    - In the Name field, specify **Sample.JMS.weather**.
    - In the JNDI name field, specify **Sample/JMS/weather**.
    - In the Topic field, specify **Sample/JMS/weather**.
11. Click **OK**.

Figure 6-151 shows the four topic resources created for the message-driven beans sample.

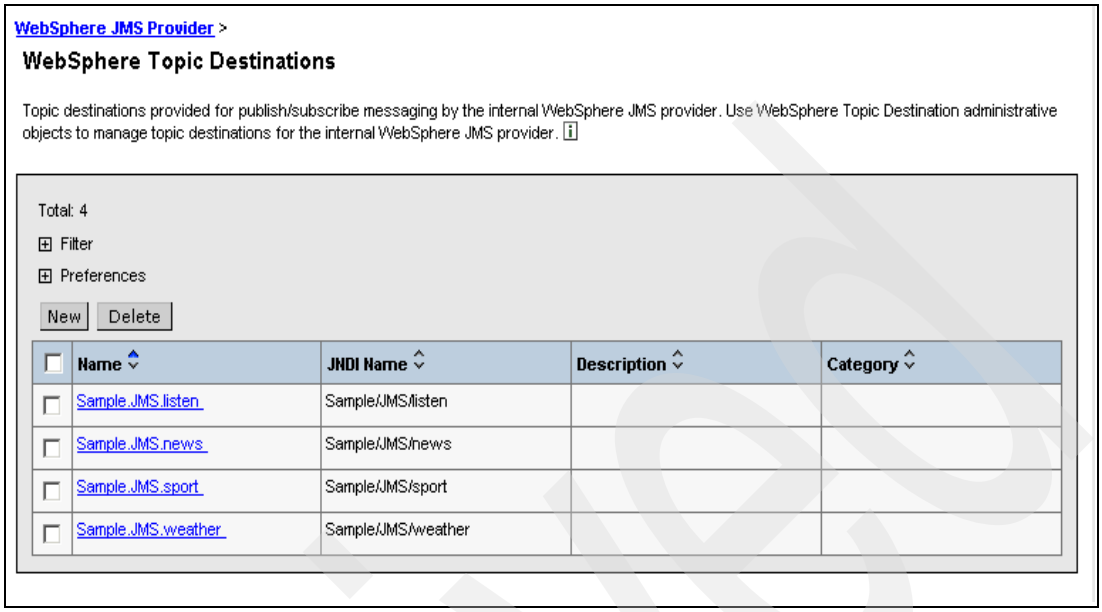


Figure 6-151 Topic destinations for message-driven bean sample

Configuring your JMS listener port

This service provides the Message Driven Bean (MDB) listening process, whereby MDBs are deployed against ListenerPorts that define the JMS destination to listen upon.

1. In the topology tree, expand **Servers**, and click **Application Servers**.
2. Click your application server; in our case, **jmsmdb**.
3. Scroll down to the bottom of the page, and click **Message Listener Service**; see Figure 6-152.

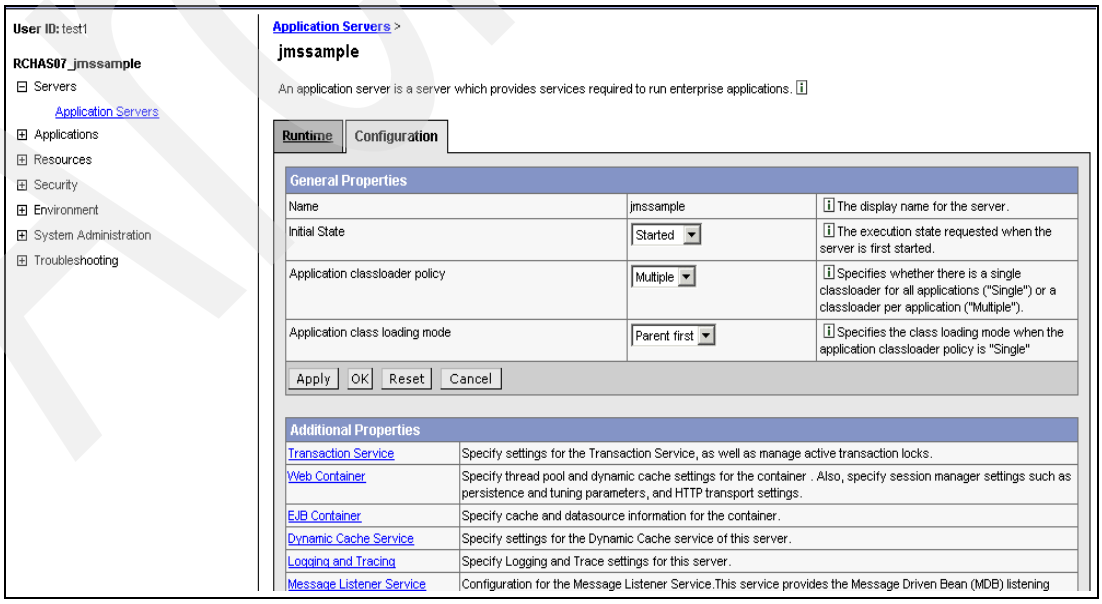


Figure 6-152 Define Message Listener Service

4. Click **Listener Ports**.
5. Click **New**; see Figure 6-153.

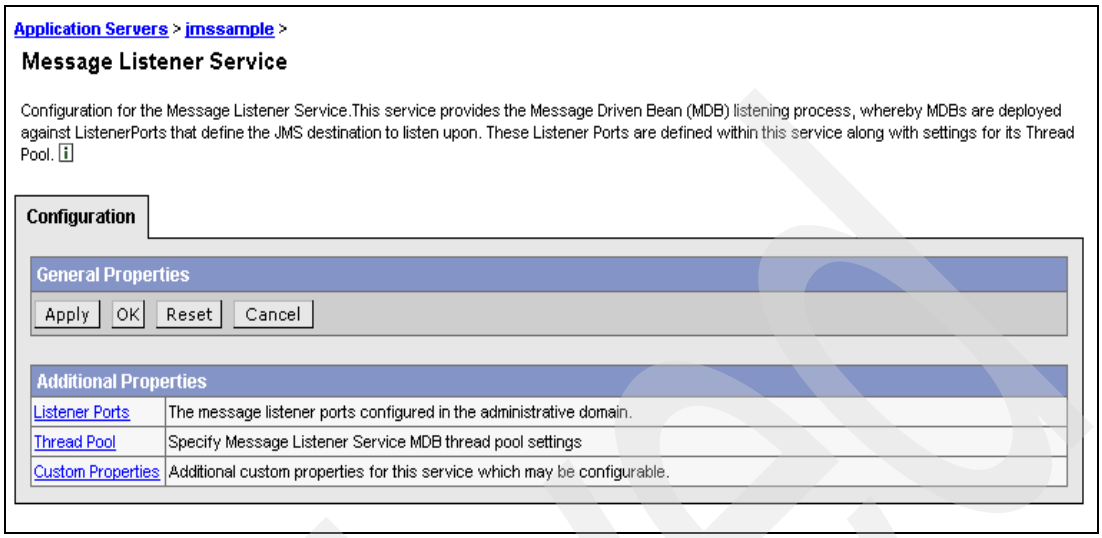


Figure 6-153 Message Listener Service

6. In the Name field, specify **SamplePubSubListenerPort**.
   - In the ConnectionFactory JNDI name field, specify **Sample/JMS/TCF**. This is the JNDI name you specified for the topic connection factory in step 4 on page 229.
   - In the Destination JNDI name field, specify **Sample/JMS/listen**. This is the JNDI name you specified for the topic destination in step 4 on page 231.
   - In the Maximum sessions field, specify **5**.
   - In the Maximum retries field, specify **2**.
   - In the Maximum messages field, specify **1**.
7. Click **OK**; see Figure 6-154.

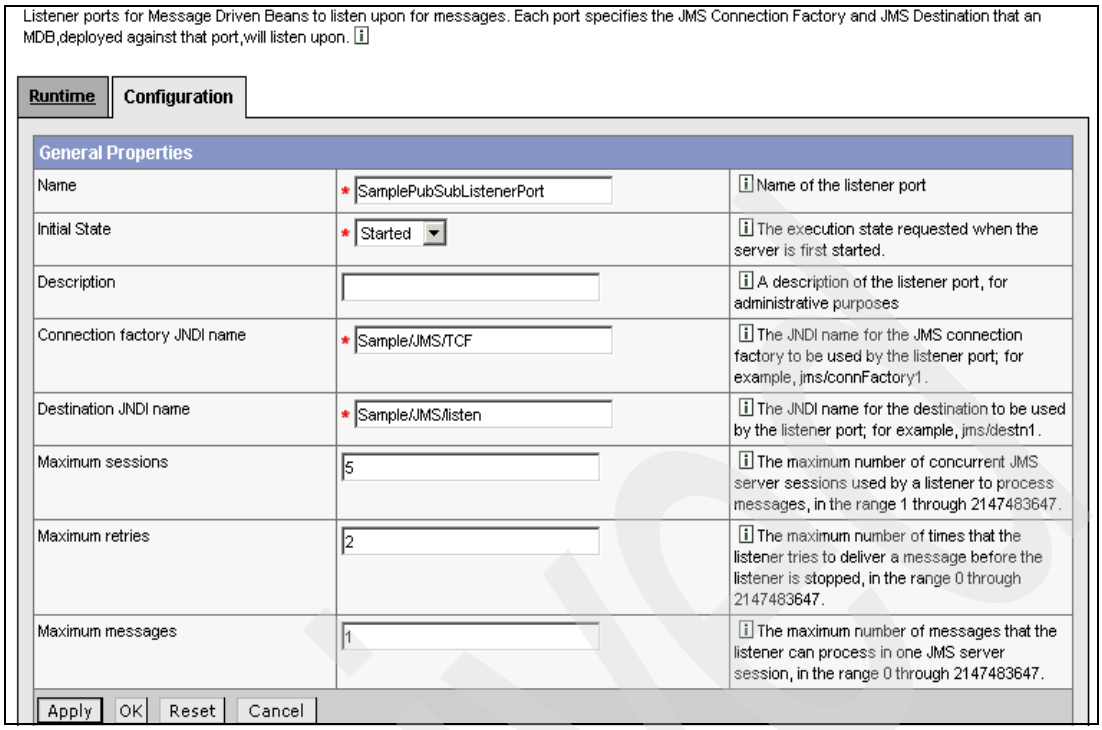


Figure 6-154 Listener ports for MDBs

8. Now all components are defined (see Figure 6-155 on page 235). Save your administrative configuration:
   a. In the Messages box at the top of the page, click **Save**.
   b. Click **Save** to save your changes to the master configuration.
   c. Click **Logout** on the toolbar, and close your browser.

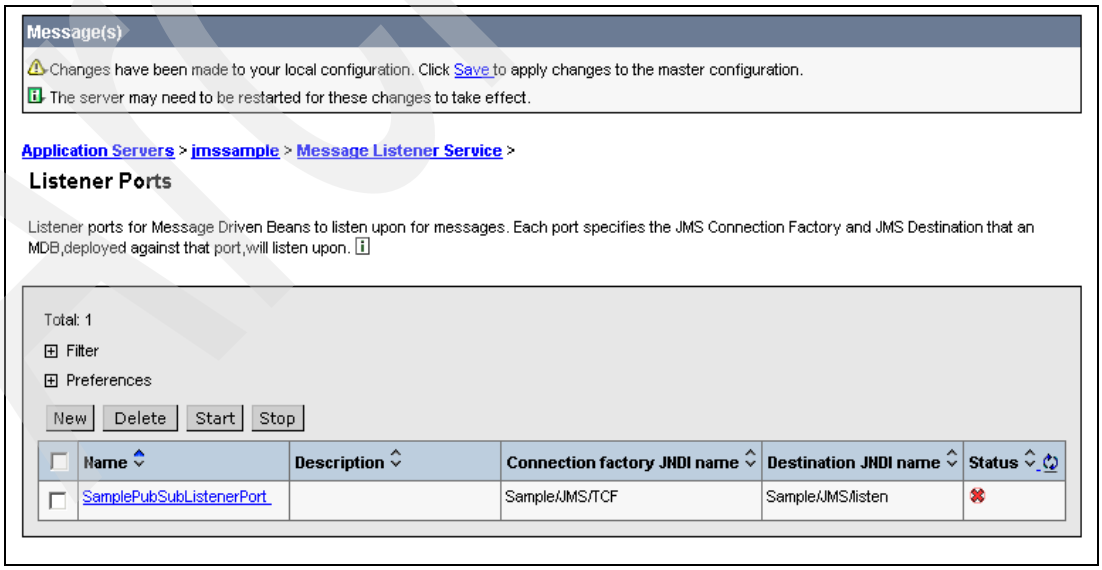


Figure 6-155 Listener ports

9. Restart your application server instance (see “Restarting the WebSphere Application Server” on page 176).

Assembling the application

Before you install the application into WAS, you need to do either of the following:

- Assemble the application by following the instructions in Information Center at:

  `http://publib.boulder.ibm.com/iseries/v1r1m0/websphere/ic2924/info/rzaiz/50/admin/bcassemmq.htm`

- Download the assembled application according to the instructions in Appendix B, “Additional material” on page 551.

Deploying your MDB application

After you create the EAR file and set up your Java Message Service (JMS) resources, the next step is to install the EAR file into your WebSphere Application Server instance. To deploy the EAR file into your application server runtime, use the Application Server administrative console:

1. Start your WebSphere Application Server instance.
2. Start the administrative console.
3. In the topology tree, expand **Applications**, and click **Install New Application**.
4. Click the **Local path** radio button and click **Browse**.

   Search for the directory and the file name of the Deployed_MDBSamples.ear file which you use to deploy the message driven bean sample into you instance jmsmdb; see Figure 6-156.

   In our case it is: `H:/QIBM/UserData/JMSSampleLab/Deployed_MDBSamples.ear`, where `H` is the network drive we assigned for our iSeries.

5. Click **Open**; see Figure 6-157.

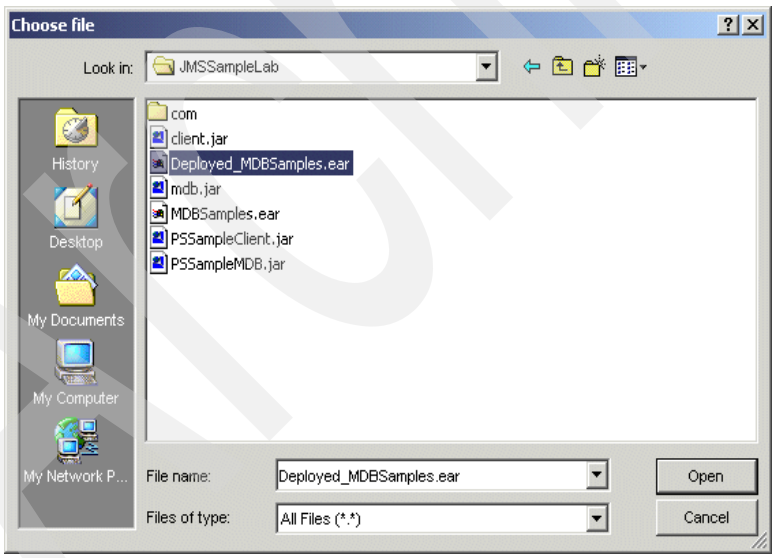


Figure 6-156 Define deployed ear file

6. Back in the Preparing for the application installation screen, click **Next**; see Figure 6-157.

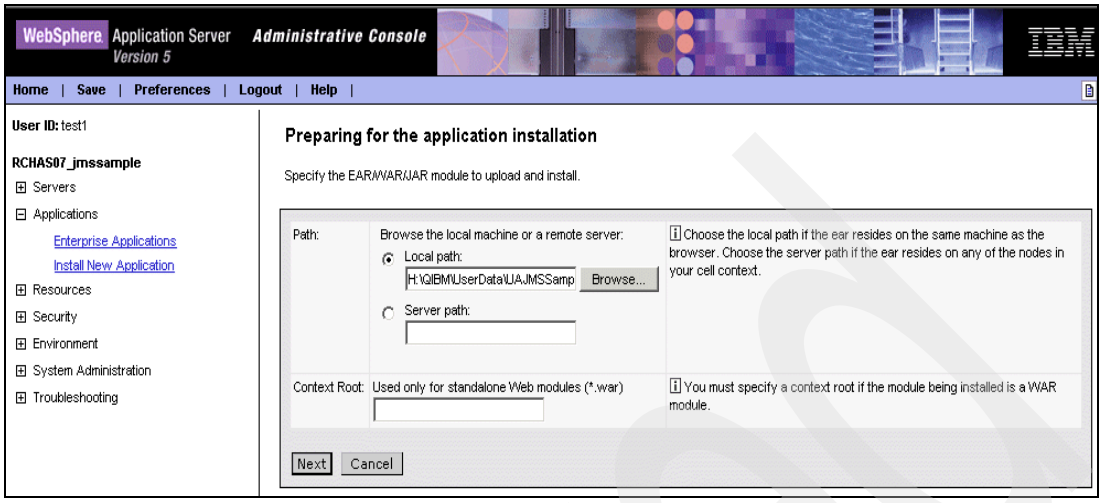


Figure 6-157 Install new application

7. Make sure that the **Do not override existing bindings** setting is checked.
8. Click **Next**; see Figure 6-158.

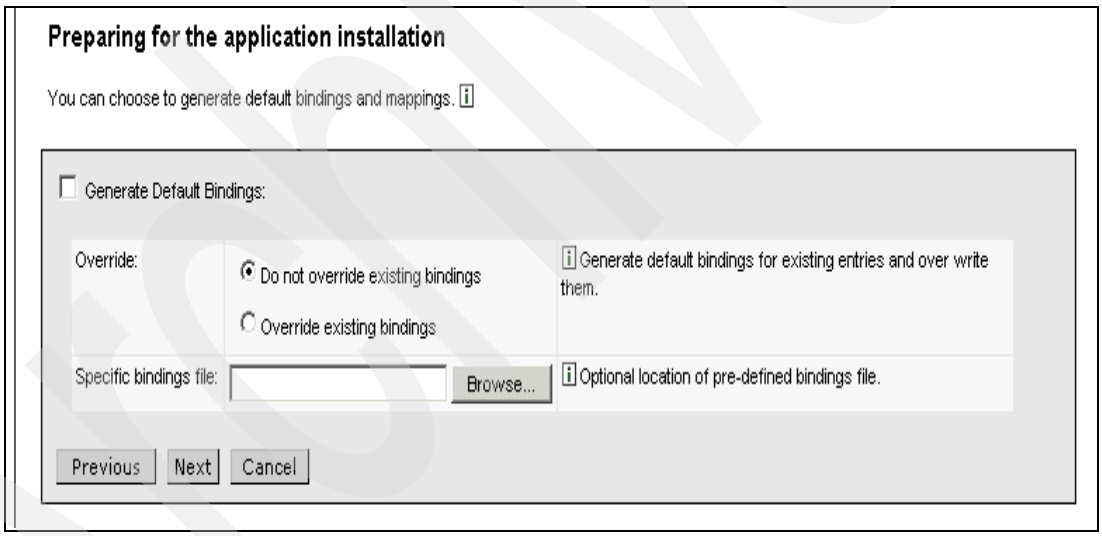


Figure 6-158 Preparing installation - binding

9. On the Step 1 page, accept all default values, and click **Next**; see Figure 6-159.

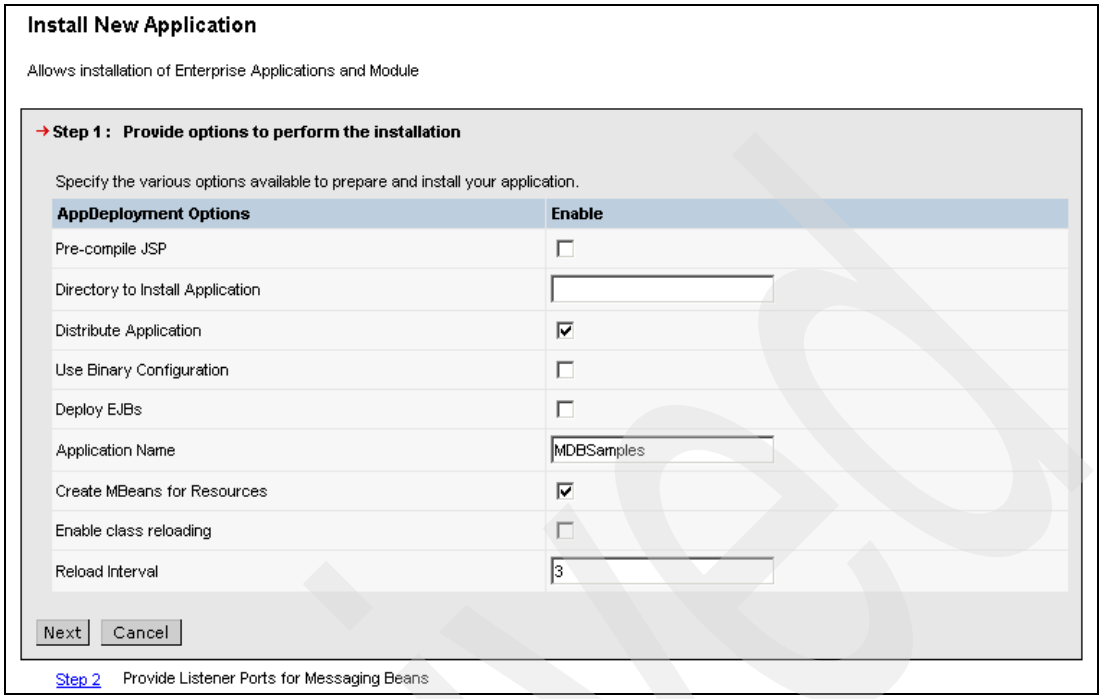


Figure 6-159 Install new application - Step 1

10. On the Step 2 page, accept all default values, and click **Next**; see Figure 6-160.

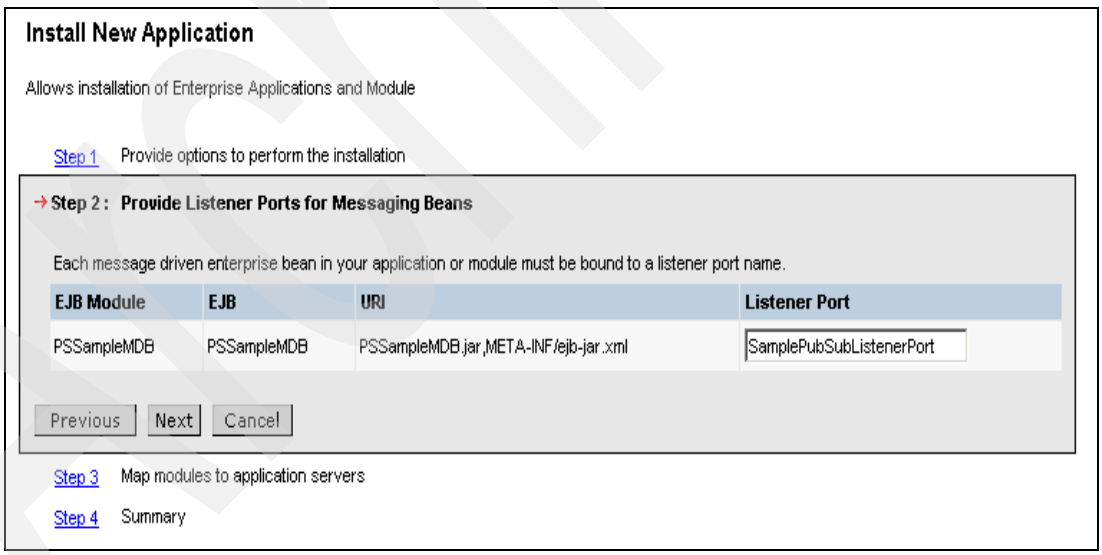


Figure 6-160 Install new application - Step 2

11. On the Step 3 page, click **Next**; see Figure 6-161.

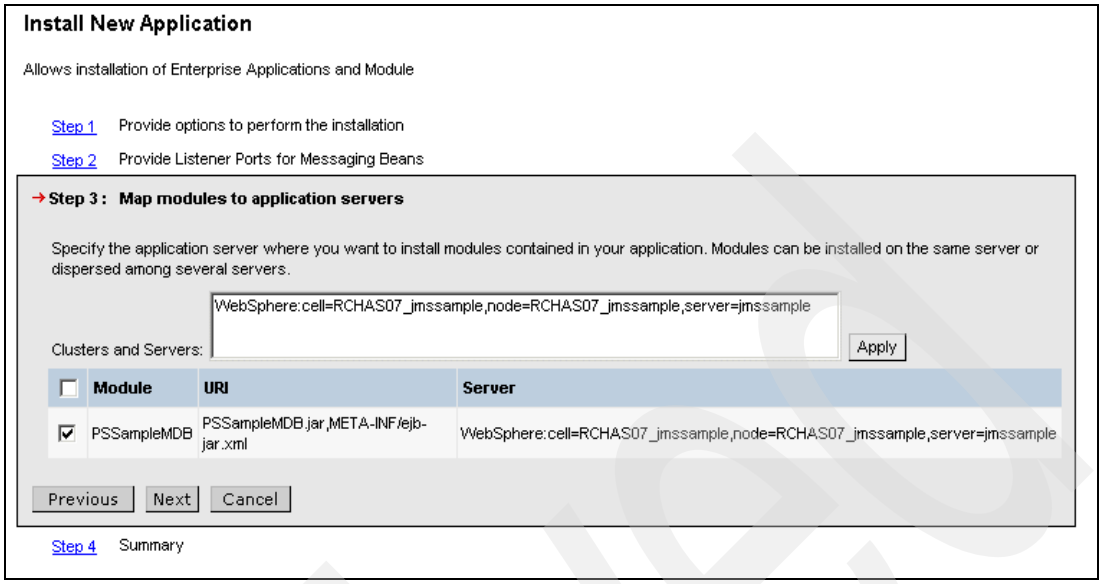


Figure 6-161 Install new application - Step 3

12. On the Step 4 page, click **Finish**; see Figure 6-162.

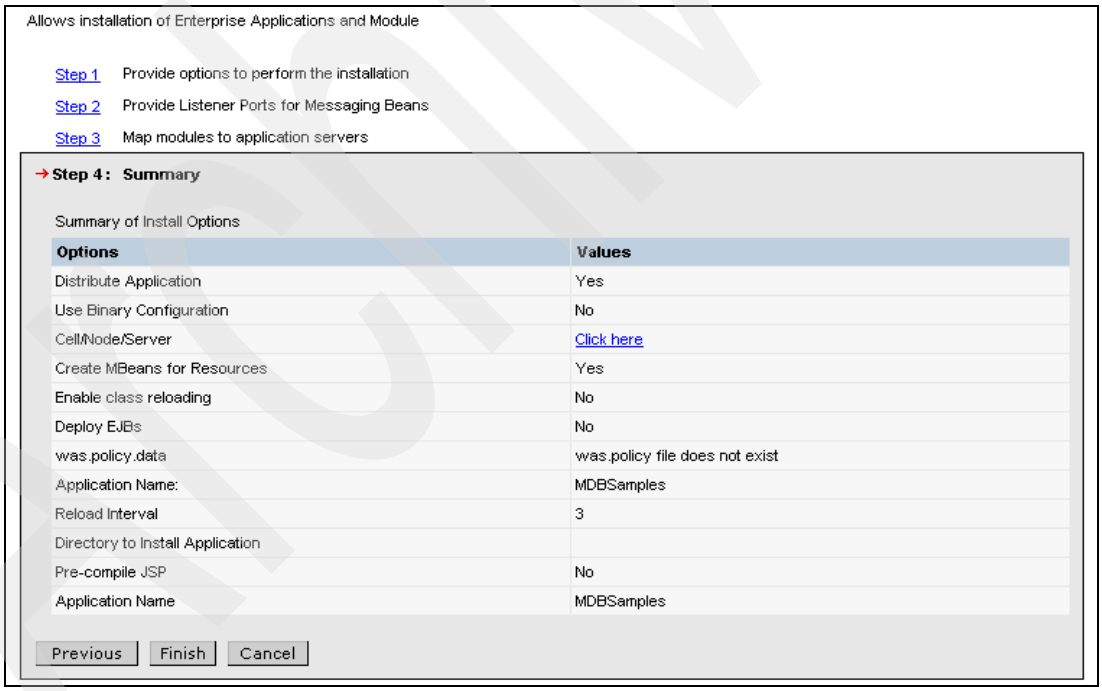


Figure 6-162 Install new application - Step 4

13. You will get a panel similar to Figure 6-163.
14. Click link **Save to Master Configuration**.

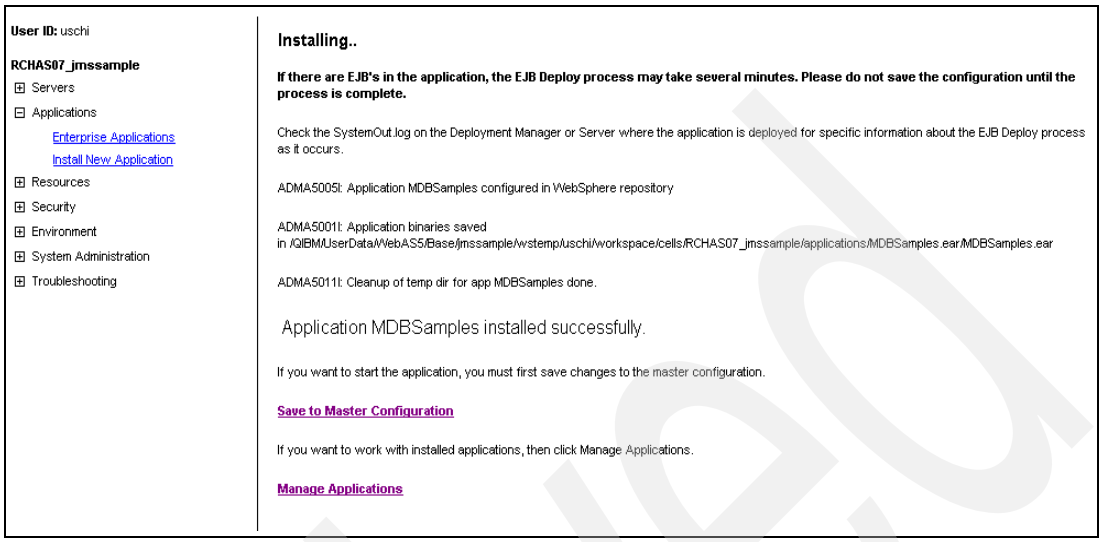


Figure 6-163 Install new application - confirmation

15. To see which items are changed, expand **View items with changes**; see Figure 6-164.
16. Click **Save**.

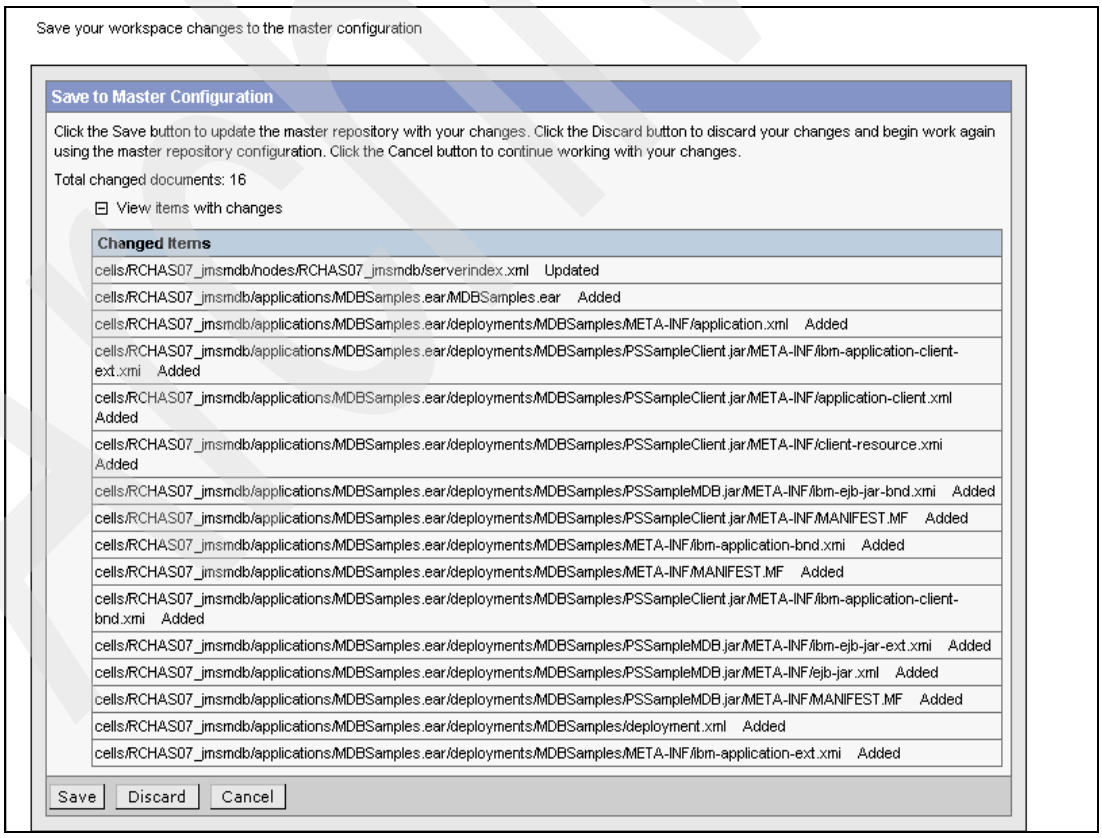


Figure 6-164 View changes items

Figure 6-165 shows the directory structure of the **jmsmdb** instance after the deployment of the new MDBSample application. Notice the new **MDBSamples.ear** directory.

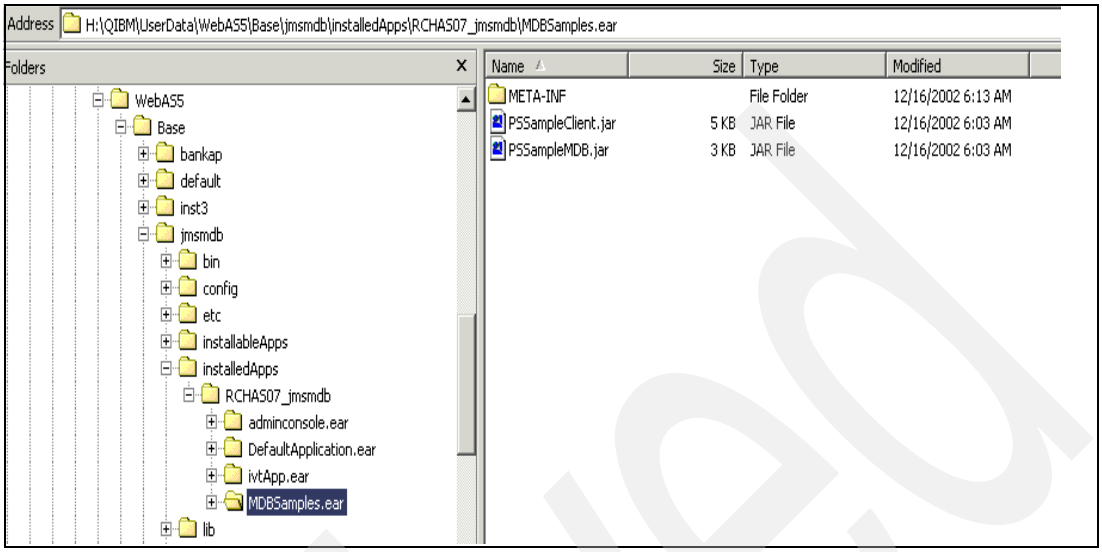


Figure 6-165 Directory structure - instance jmsmdb after deployment

Starting your new MDB application

Follow these steps to start your MDB application:

1. In the topology tree of the administrative console, expand **Applications**, and click **Enterprise Applications**; see Figure 6-166.

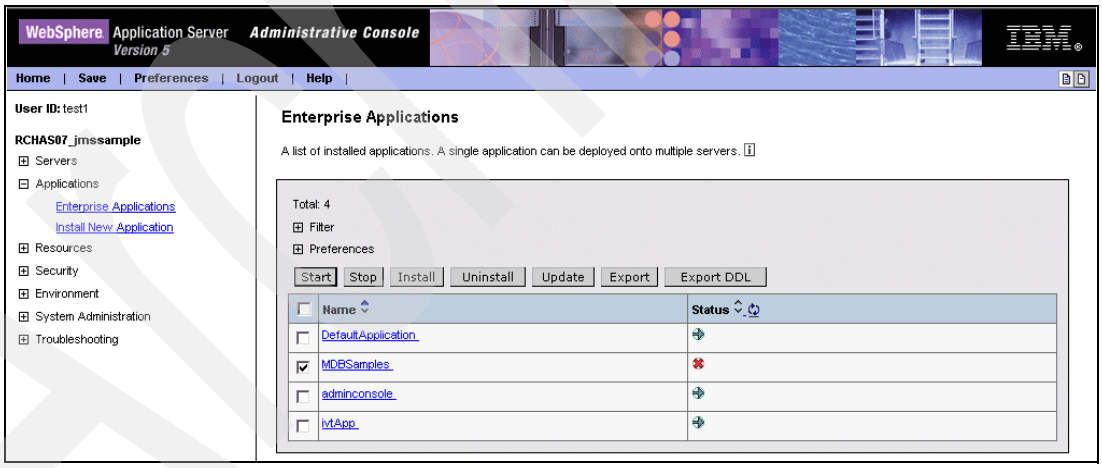


Figure 6-166 Installed enterprise applications

2. Select **MDBSamples**, and click **Start**.
3. You will receive a message that the application started successfully; see Figure 6-167.

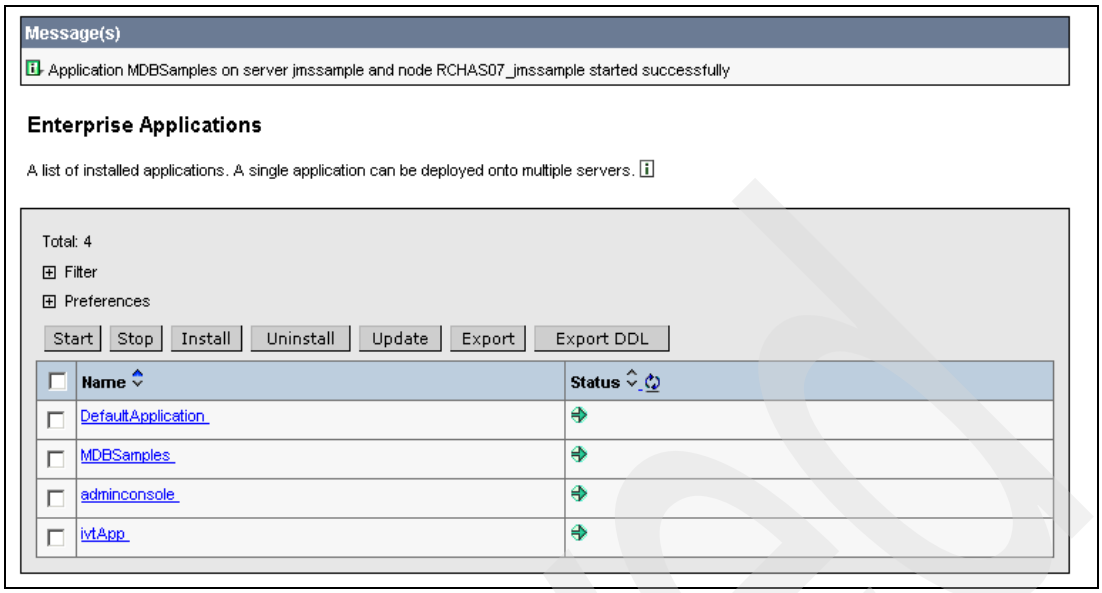


Figure 6-167 Started new MDB Samples application

Running the new MDB application

After the successful deployment of the message-driven beans sample application to your WebSphere Application Server instance, place a copy of the installed EAR file in your installableApps folder, and run the application on your iSeries server.

1. Enter the Start QShell (STRQSH) command on an OS/400 command line.

   To locate the installed EAR file, enter this command:

   ```
   cd /QIBM/UserData/WebAS5/Base/instance_name/installedApps/hostName_instance_name
   ```

   Here, `hostName` is the host name of your iSeries server, and `instance_name` is the name of your instance. In our case it is:

   ```
   cd /QIBM/UserData/WebAS5/Base/jmsmdb/installedApps/RCHAS07_jmsmdb
   ```

2. Copy the message-driven beans sample application EAR file (MDBSamples.EAR) to your installableApps directory. Enter this command:

   ```
   cp -R MDBSamples.ear /QIBM/UserData/WebAS5/Base/instance_name/installableApps
   ```

   In our case:

   ```
   cp -R MDBSamples.ear /QIBM/UserData/WebAS5/Base/jmsmdb/installableApps
   ```

The successful deployment of the EAR file must be tested to make sure that all JMS resources have been correctly set up. Perform the following steps to run the application client on the iSeries server:

1. Enter the Start QShell (STRQSH) command on an OS/400 command line.
2. Enter the following command to run the application client:

   ```
   /QIBM/ProdData/WebAS5/Base/bin/launchClient -instance instance_name
   /qibm/userdata/webas5/base/instance_name/installableApps/MDBSamples.ear
   -CCjar=PSSampleClient.jar -CCBootstrapPort=your_Name service port
   -CCsoapConnectorPort=your_soap_port -verbose -topic news -msg
   "any Message Text"
   ```

   Here, `instance_name` is the name of your WAS instance, `your_Name service port` is the Name Service port of your WAS application server, and `your_soap_port` is the SOAP port of your WAS application server.

Note: The *bootstrap port*, as it is often referred to in the WAS documentation, is the same as *Name service port* of your instance. Port numbers are assigned when you create your instance. To display the port numbers assigned to your instance, you can use the `dspwasinst` script.

Figure 6-168 shows the ports of our instance jmsmdb.

```
dspwasinst -instance jmsmdb
ADCP0005I: Using cell RCHAS07_jmsmdb, node RCHAS07_jmsmdb and server *ALL.
Display WAS instance:
  Instance name: jmsmdb
  Instance type: Base Application Server
  Cell: RCHAS07_jmsmdb
  Node: RCHAS07_jmsmdb

  Information for server: jmsmdb
    Installed applications:
      DefaultApplication
      ivtApp
      adminconsole
      MDBSamples

    Ports in use:
         10402   Administrative console port
         10414   Administrative console SSL-enabled port
         10405   Name service port
         10410   Soap port
         10401   Internal HTTP port
         10409   Data replication service client port
         10406   Internal Java Message Service server port
         10407   Queued Java Message Service server port
         10408   Direct Java Message Service server port
         10411   SAS SSL server authentication listener port
         10413   CSIV2 server authentication listener port
         10412   CSIV2 mutual authentication listener port

$
```

Figure 6-168 Display instance jmsmdb

In our case, we type in the following command to run the application client and sent a message to the `sport` topic destination. You can also sent the message to the `news` or `weather` topic destination by defining it with the `-topic` parameter:

```
/QIBM/ProdData/WebAS5/Base/bin/launchClient -instance jmsmdb
/qibm/userdata/webas5/base/jmsmdb/installableApps/MDBSamples.ear
-CCjar=PSSampleClient.jar -CCBootstrapPort=10405
-CCsoapConnectorPort=10410 -verbose -topic sport -msg "This is Uschi crying for help, I
want to have snow"
```

The JMS client sends the message as described above and reports the progress. The output is similar to that shown in Figure 6-169.

```
/qibm/proddata/webas5/base/bin/launchClient -instance jmsmdb
/qibm/userdata/webas5/base/jmsmdb/installableApps/MDBSamples.ear -CC
jar=PSSampleClient.jar -CCBootstrapPort=10405 -CCsoapConnectorPort=10410 -verbose -topic
sport -msg "This is Uschi crying for help, I want to have snow"

IBM WebSphere Application Server, Release 5.0
J2EE Application Client Tool
Copyright IBM Corp., 1997-2002
WSCL0012I: Processing command line arguments.
WSCL0013I: Initializing the J2EE Application Client Environment.
WSCL0035I: Initialization of the J2EE Application Client Environment has completed.
WSCL0014I: Invoking the Application Client class
com.ibm.websphere.samples.messaging.pubsub.JMSpsSampleClient Topic:sport.
 Sending message: 'This is Uschi crying for help, I want to have snow'
 Retrieving a TopicConnectionFactory from JNDI
 Retrieving Topic from JNDI
 Creating a Connection
 Starting the Connection
 Creating a Session
 Creating a TopicPublisher
 Creating a TextMessage
 Publish the message to topic://Sample/JMS/sport?brokerVersion=1
 Closing Publisher
 Closing session
 Closing connection
 End of Sample
 $
```

Figure 6-169 Client application sends a message

6.13.4 Modifying JMS resources

To modify a JMS resource, follow these steps:

1. Start the administrative console.
2. In the topology tree, expand **Resources** and click **WebSphere JMS Provider**.
3. On the WebSphere JMS Provider page, click the link for the type of resource that you want to modify:
   - WebSphere Queue Connection Factories
   - WebSphere Topic Connection Factories
   - WebSphere Queue Destinations
   - WebSphere Topic Destinations
4. Click the **name of the resource** that you want to modify.
5. Make your changes and click **OK**.
6. **Save** the configuration.

6.13.5 Removing JMS resources

To remove a JMS resource, follow these steps:

1. Start the administrative console.
2. In the topology tree, expand **Resources** and click **WebSphere JMS Provider**.

3. On the WebSphere JMS Provider page, click the link for the type of resource that you want to remove:
   - WebSphere Queue Connection Factories
   - WebSphere Topic Connection Factories
   - WebSphere Queue Destinations
   - WebSphere Topic Destinations
4. On the page for the type of resource, **select the checkbox** for the resource that you want to remove; see Figure 6-170.

> **Note:** You can select more than one resources for delete operation in one step.

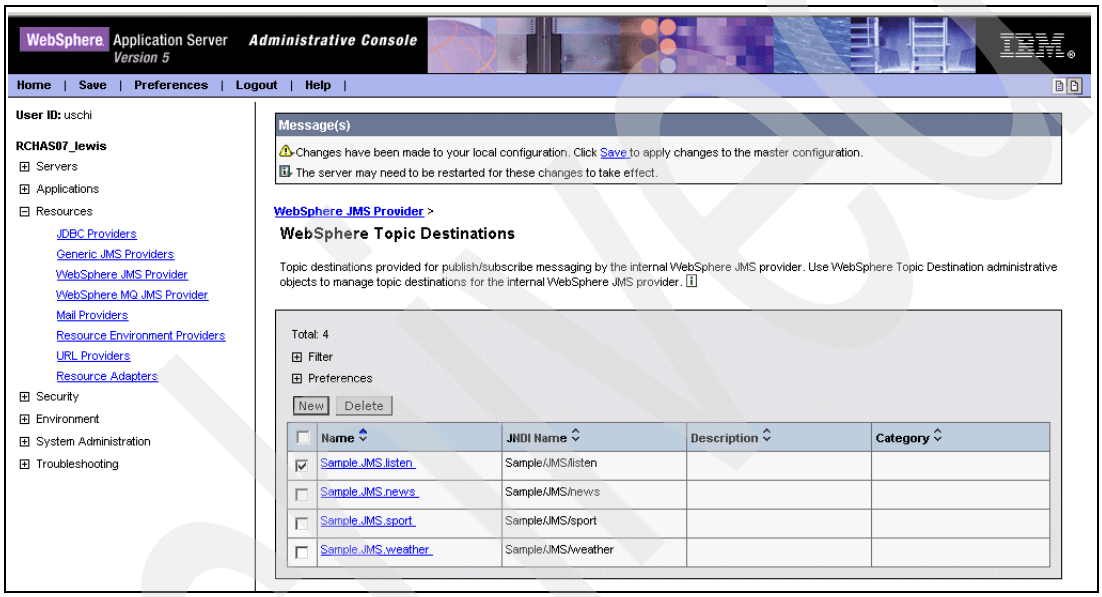


Figure 6-170 JMS Resources for instance lewis - before delete

5. Click **Delete**.
6. **Save** the configuration.

6.13.6 Installing a sample WebFacing application

In this section we show how to install an application that is build with the IBM WebFacing Tool.

> **Note:** This application is not included as part of our additional materials.

The IBM WebFacing Tool allows to Web-enable a RPG or COBOL program. These programs continue to run “as-is”, but the green-screen is replaced with a browser based front-end. This is called “Web faced”, changing the face of the application. The goal of WebFacing for iSeries is to quickly convert the appearance of the existing interactive applications into browser-based GUIs. This is done with minimal change to underlying applications and minimal initial investment in skills.

The WebFacing Tool uses the display file, the DDS (Data Description Specifications) file that is linked to a program, and converts this file to JavaServer Pages (JSPs) and JavaBeans. The application program remains intact and the display portion of the application is now “Web enabled”. The DDS to JSP conversion takes place in the WebFacing Tool wizard, which is a Windows based tool that is now included in WebSphere Development Studio Client for iSeries. One important benefit of using the WebFacing Tool is that all the components, JSPs and JavaBeans, are created at development time. For more information on the WebFacing Tool, visit the following Web site:

`http://ibm.com/software/ad/wdt400/webfacing/`

A WebFaced application is packaged in an EAR file and deployed to WAS.

Here we show the steps to install an sample application that is build by the WebFacing Tool. We install this application in our WEBFACING instance:

1. Be sure that your WAS instance is started.
2. Start the administrative console.
3. Expand **Applications** and select **Install New Application**.
4. On the Preparing for application installation select **Local path** and click **Browse**.

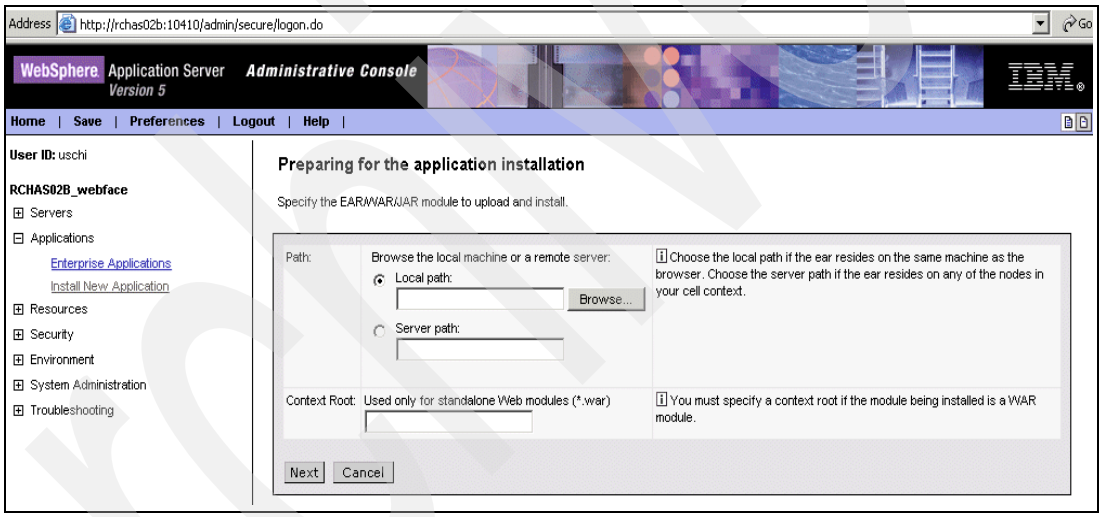


Figure 6-171 Preparing for the application installation

5. Navigate to the EAR file and click **Open**.

   In our case it is the **webfacing.ear** file (seeFigure 6-172).

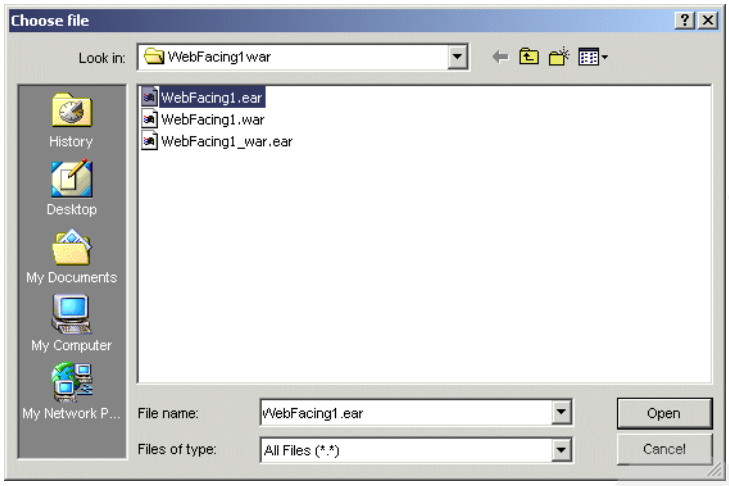


Figure 6-172 select webfacing.ear file

6. Back in the Preparing for the application installation panel, click **Next** (see Figure 6-173).

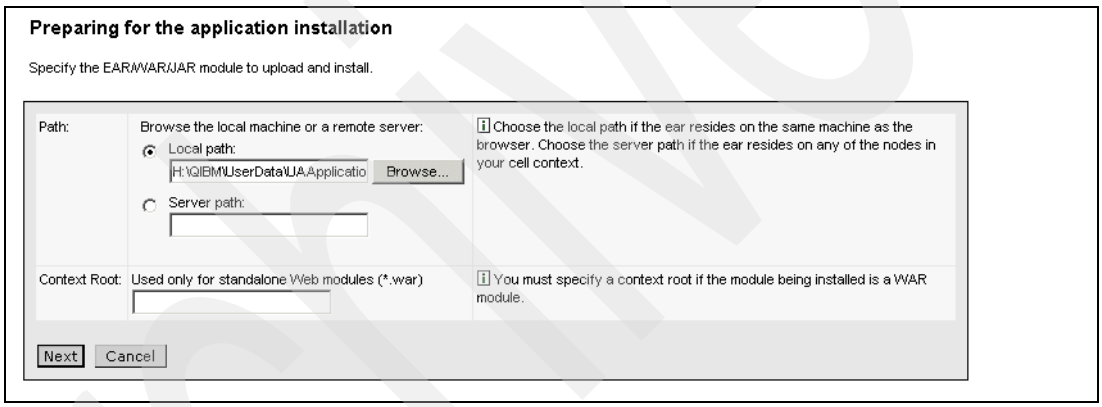


Figure 6-173 Preparing for application installation

7. In the next panel,accept the default values and click **Next** (see Figure 6-174).

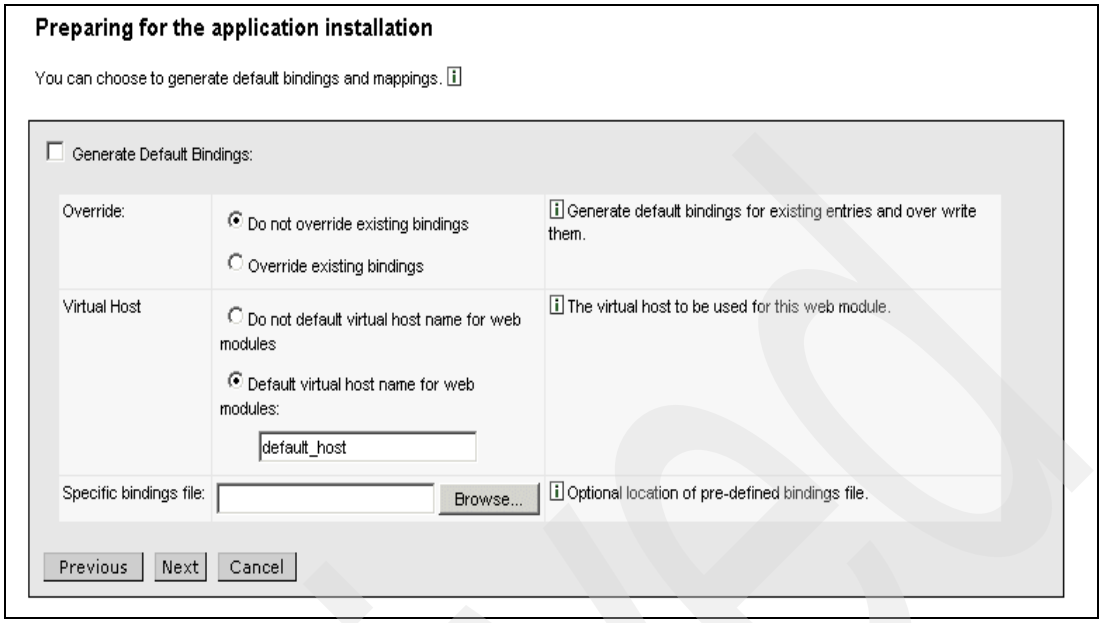


Figure 6-174 Set bindings and define virtual host name

8. In the step 1 panel change the Application Name to **WebFacing1**, as shown in Figure 6-175 and click **Next**.

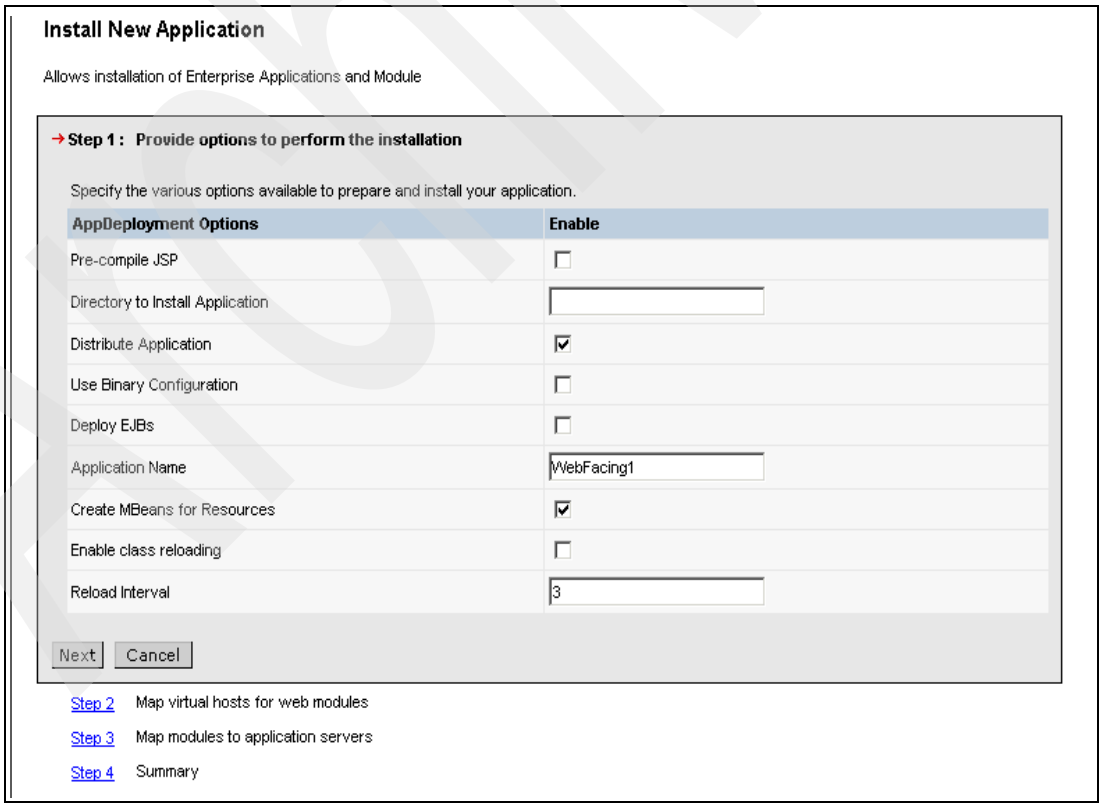


Figure 6-175 Step1 panel

9. Click **Step 4**.

10. In the Summary panel, as shown in Figure 6-176 and Figure 6-177, click **Finish**.

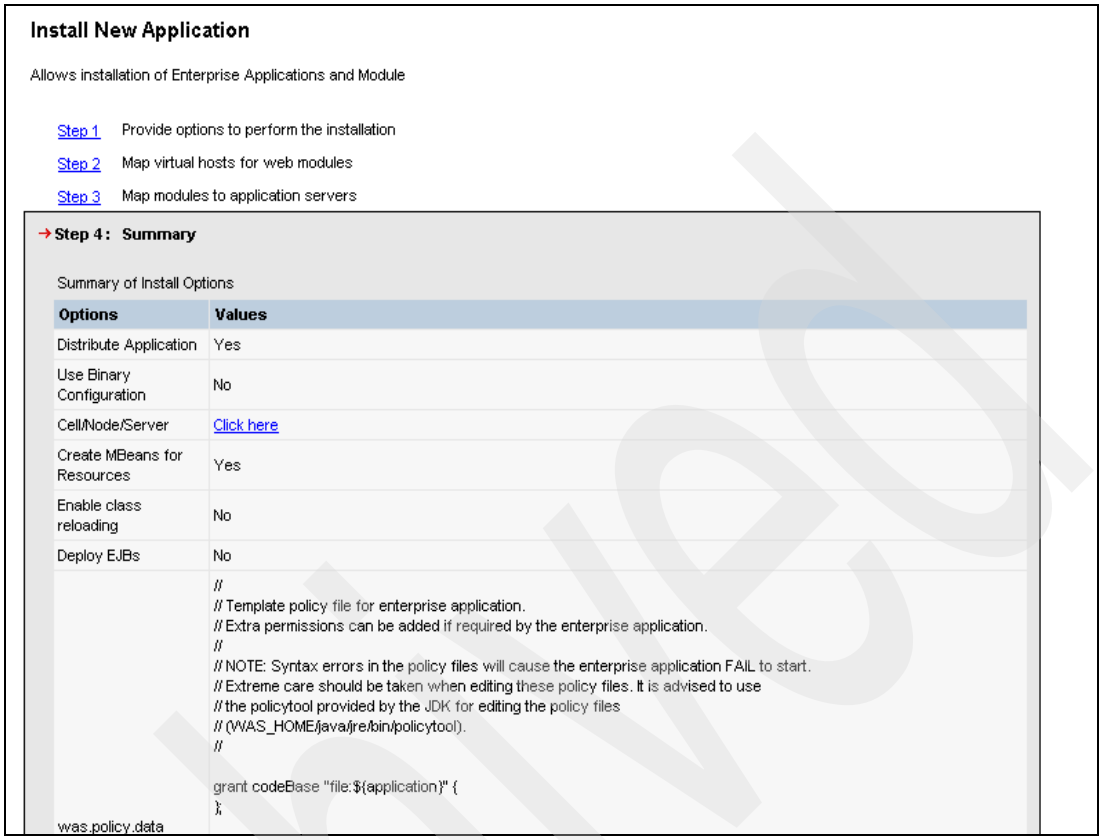


Figure 6-176 Summary panel, part 1 of 2

Figure 6-177 Summary panel, part 2 of 2

11. You will see messages in the administrative console as shown in Figure 6-178.

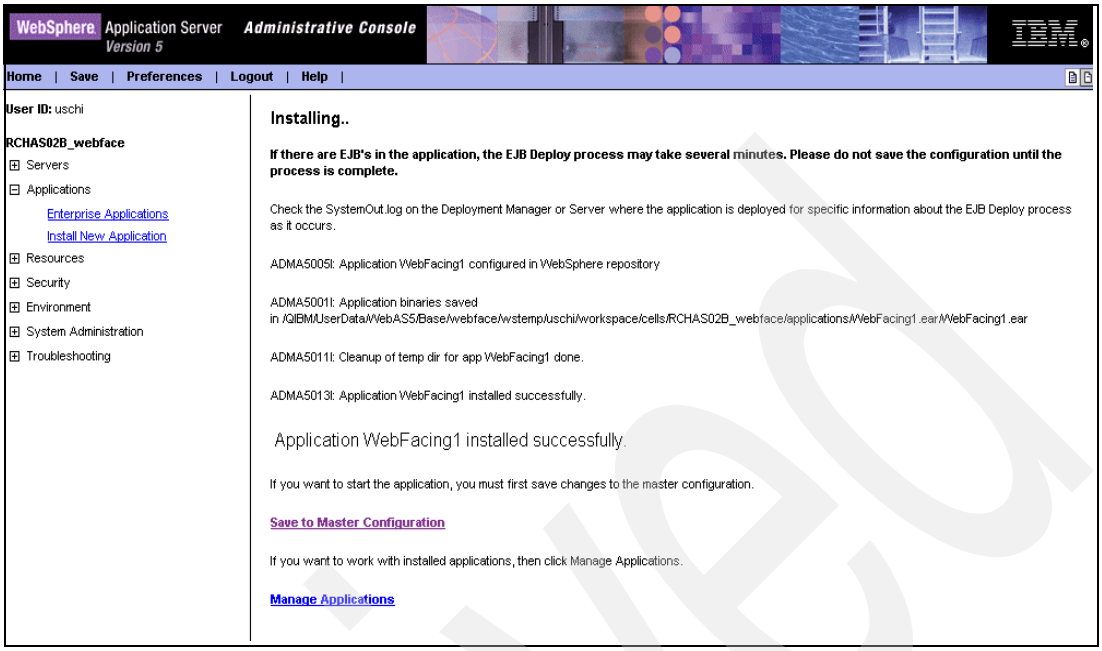


Figure 6-178 Installation messages in administrative console

12. Save the configuration changes by clicking **Save to Master Configuration**.
13. When the Save to master configuration panel appears, click **Save**.
14. Because you installed a new application with Web module, perform the **Update web server plugin**; see 6.7.4, “Updating Web server plug-in configuration” on page 171.
15. Start the new WebFacing1 application via the administrative console as described in “Starting your new MDB application” on page 241.
16. Start the WebFacing server (part of the WebFacing Tool) via an iSeries command line. Type the following:

```
strtcpsvr *webfacing
```

17. Test your application using a browser. In our case we use the URL:

```
http://rchas02b:10416/WebFacing1.
```

Figure 6-180 shows the 5250 interface to the application.

The original 5250 RPG application is still used (is invoked and running), but all communication between the user and the program are done via the browser, which uses the new components (JSPs and servlets) that are created by the WebFacing tool.

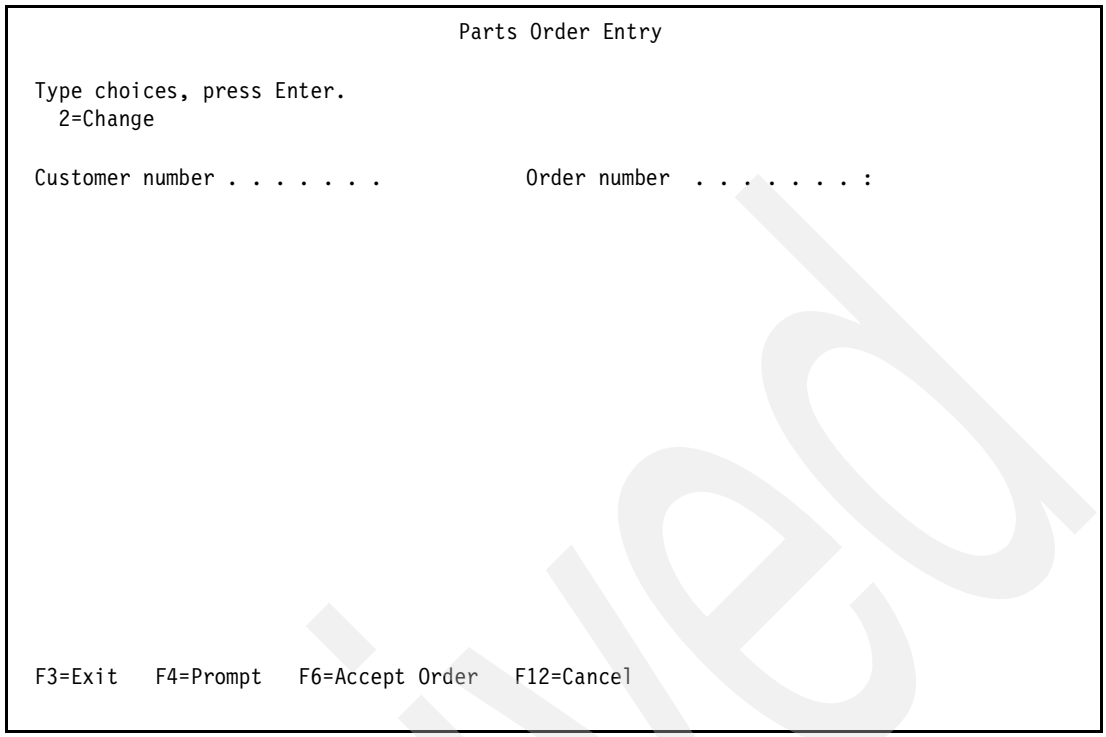


Figure 6-179 First screen of original 5250 Order Entry Application

The first screen (and all others) of our 5250 Order Entry application comes now with the new look via the WebFacing1 application, like that shown in Figure 6-180.

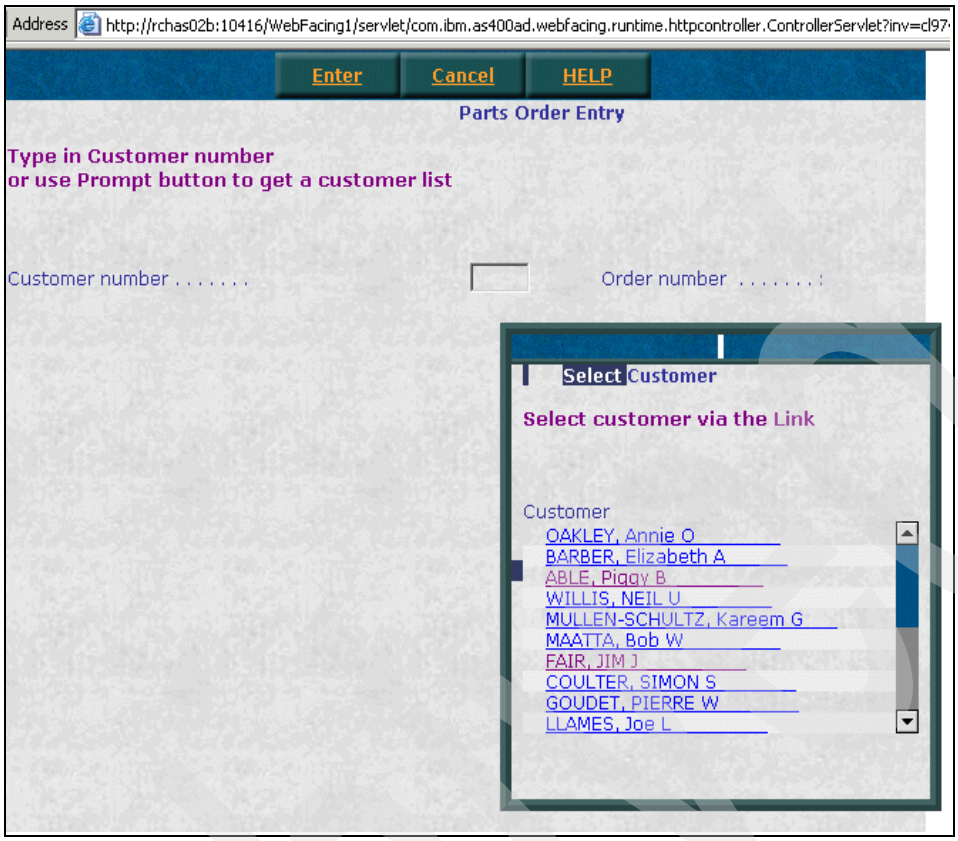


Figure 6-180 First screen of WebFacing1 order entry application

6.13.7 Uninstalling an application

To uninstall an application, follow these steps:

1. Start the administrative console.
2. Expand **Applications** and click **Enterprise Applications**.
3. Select the check box for the application or applications that you want to remove.
4. Stop the application you want to uninstall by clicking the **Stop** button.
5. Select the check box for the application or applications that you want to remove.
6. Click **Uninstall**; see Figure 6-181.

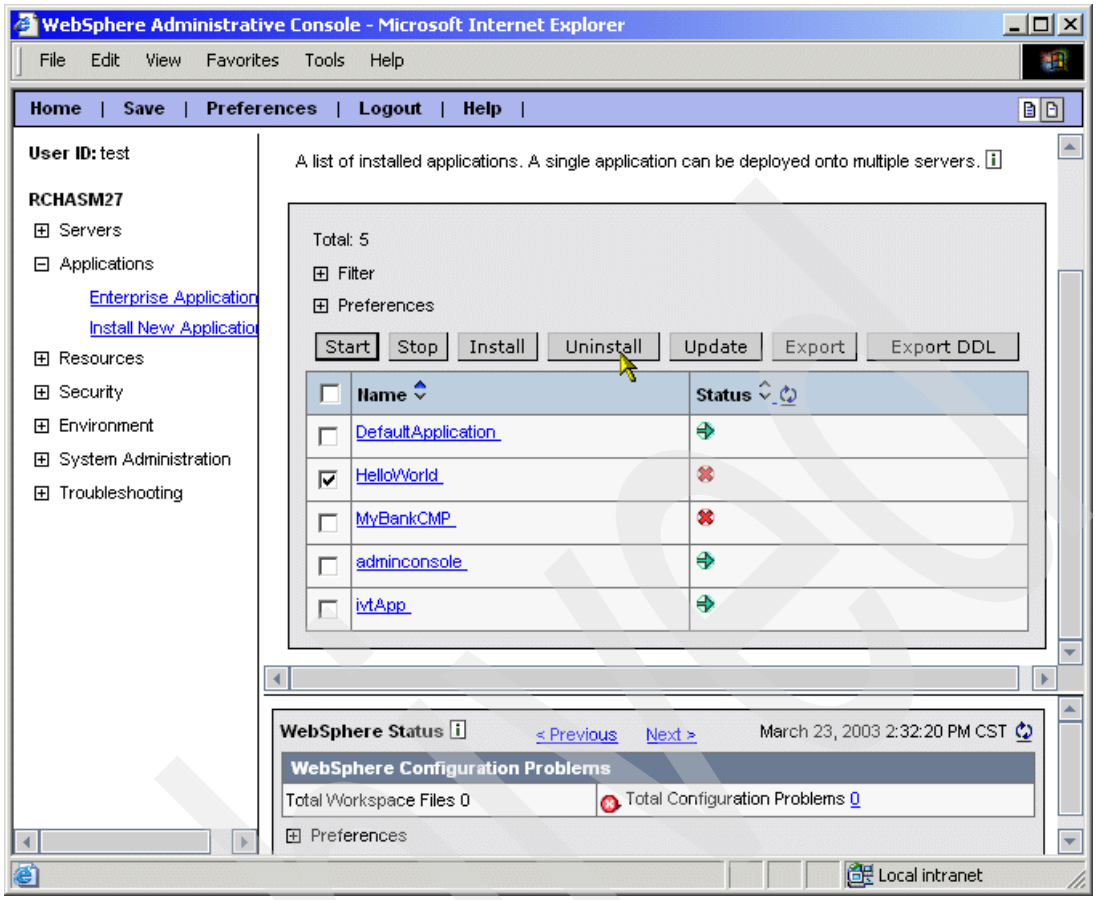


Figure 6-181 Uninstall an application

7. Save the configuration.

6.14 Administering shared libraries

WebSphere Application Server uses shared libraries to define code that is used by your enterprise applications, but not packaged within those applications. For example, a utility library, such as IBM Toolbox for Java classes, could be packaged as a shared library. Shared libraries can be associated with an enterprise application or an application server. If you associate a shared library with an application server, it is accessible to all of the enterprise applications deployed on that server.

Shared library files in WebSphere Application Server consist of a symbolic name, a classpath, and a path for loading JNI libraries. You can define a shared library at the cell, node, or server level. Defining a library at one of the three levels does not cause the library to be placed into the application server's classloader. To make the classes in the library available to an application or server, you must create an association between the library and the application or server.

A separate classloader is used for shared libraries that are associated with an application server. This classloader is the parent of the application classloader, and the WebSphere Application Server extensions classloader is its parent. Shared libraries that are associated with an application are loaded by the application classloader. For more information about classloaders; see the WAS online documentation center via the following Web site:

`http://publib.boulder.ibm.com/iseries/v1r1m0/websphere/ic2924/index.htm?info/rzaiz/50/was.htm.`

6.14.1 Configuring shared libraries

In this section we show how to manage shared libraries. We provide instructions on how to:

- Create shared libraries
- Modify shared libraries
- Remove shared libraries
- Associate a shared library with an application
- Associate a shared library with an application server

Creating a shared library

To create a shared library, follow these steps:

1. Start the administrative console.
2. In the topology tree, expand **Environment** and click **Shared Libraries**.
3. On the Shared Libraries page, **specify the scope** for which you want to define a library and click **Apply** (see Figure 6-182).
4. Click **New** (see Figure 6-182).

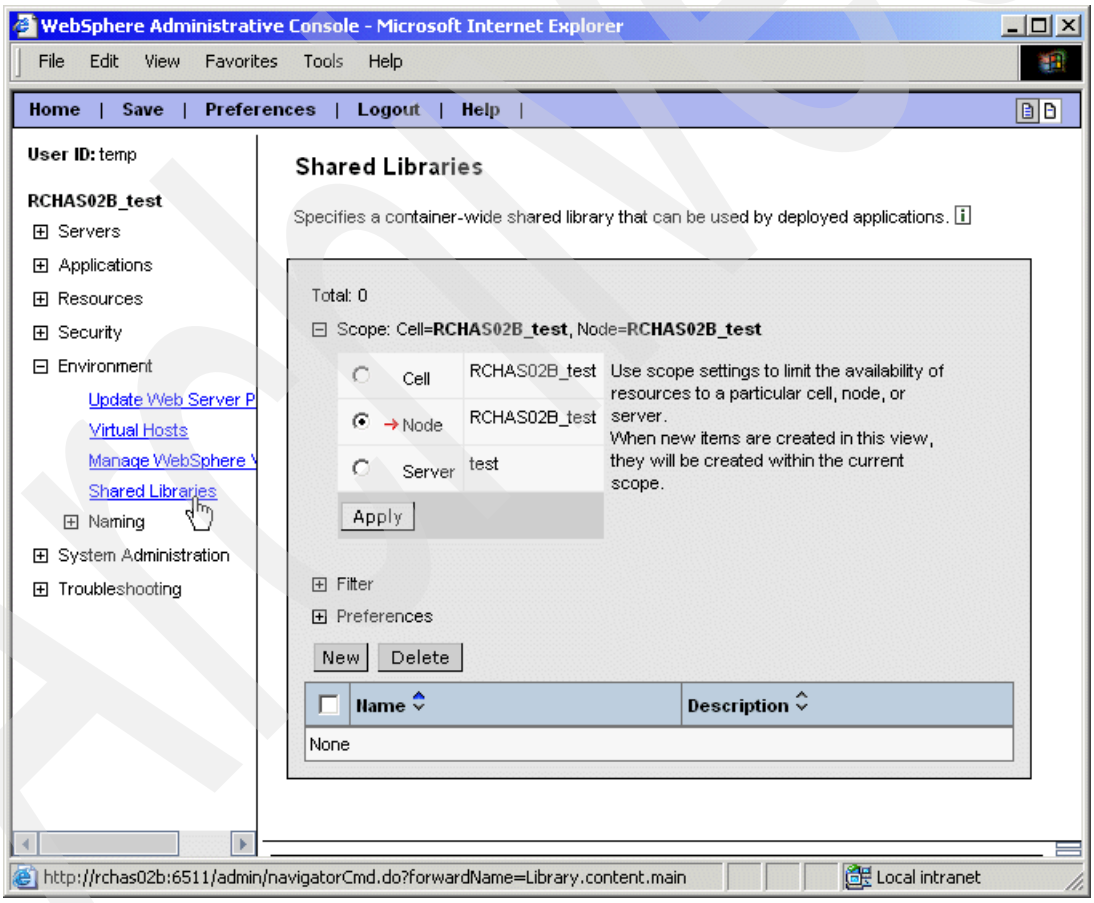


Figure 6-182 Creating a new shared library

5. In the next panel, specify a name and classpath for the shared library. You can also specify other properties on this page (see Figure 6-183). For the instruction purposes we create two shared libraries that are based on the Toolbox for Java library, `jt400.jar`, and classes for the native JDBC driver, `db2_classes.jar`.
6. Click **OK**.

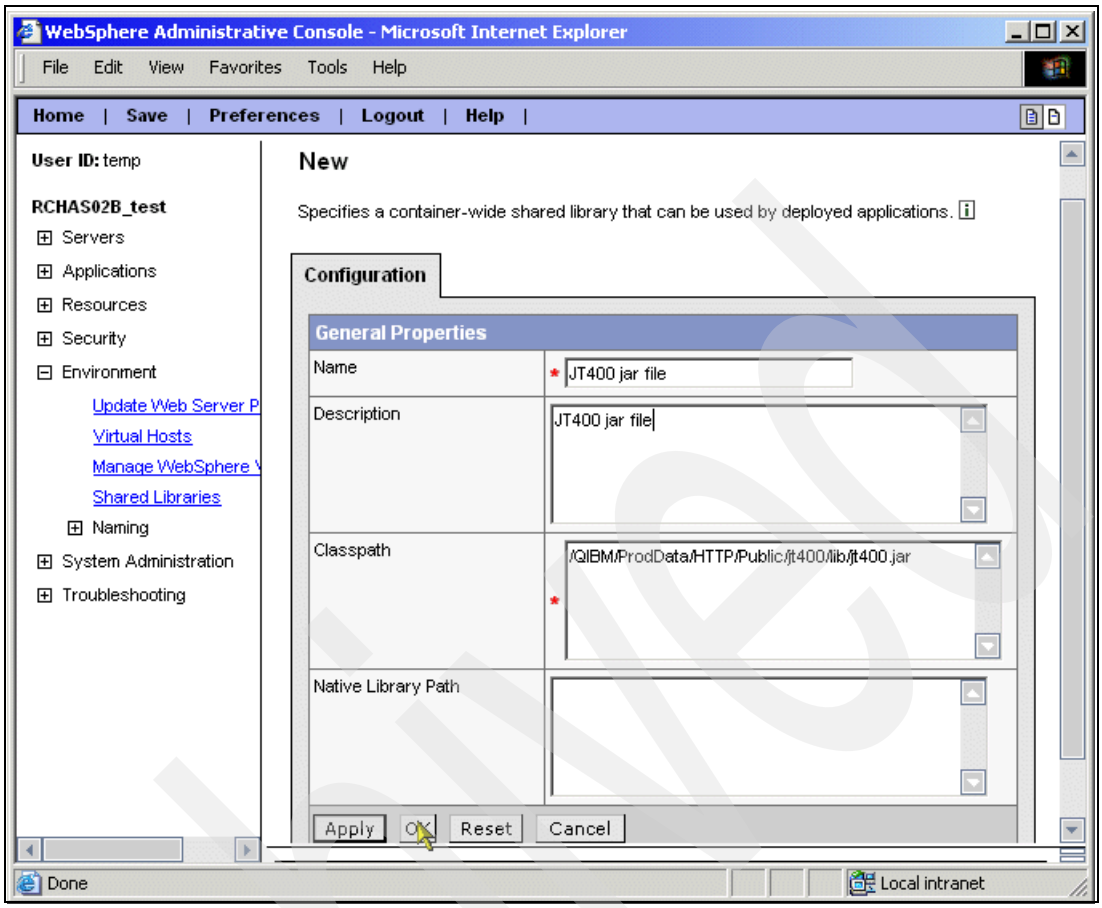


Figure 6-183 Specifying the parameters for a shared library

7. Perform the previous steps for the second shared library. For the classpath parameter, use `/QIBM/UserData/Java400/ext/db2_classes.jar`.
8. **Save** the configuration.
9. Restart either:
   - The application server, if the scope of change is at the application server level
   - The application, if the scope of change is at the application level

Modifying shared libraries

To modify a shared library, follow these steps:

1. Start the administrative console.
2. In the topology tree, expand **Environment** and click **Shared Libraries**.
3. On the Shared Libraries page, **specify the scope** contains the library that you want to modify and click **Apply**.
4. Click the name of the shared library that you want to modify.
5. Make your changes.
6. Click **OK**.
7. **Save** the configuration.
8. Restart either:
   - The application server if the scope of change is at the application server level
   - The application, if the scope of change is at the application level

Removing shared libraries

To remove a shared library, follow these steps:

1. Start the administrative console.
2. In the topology tree, expand **Environment** and click **Shared Libraries**.
3. On the Shared Libraries page, **specify the scope** for that contains the library that you want to remove and click **Apply**.
4. Select the checkbox for the shared library that you want to remove.
5. Click **Delete**.
6. **Save** the configuration.
7. Restart either:
   - The application server if the scope of change is at the application server level
   - The application, if the scope of change is at the application level

Associating a shared library with an application server

In order for the shared library to be useful, you need to associate it either with an application of application server. To associate a shared library with an application server, follow these steps:

1. Start the administrative console.
2. In the topology tree, expand **Servers** and click **Application Servers**.
3. On the Application Servers page, click the name of the server to which you want to add a shared library.
4. On the application server's detail page, click **Classloader** (see Figure 6-184).

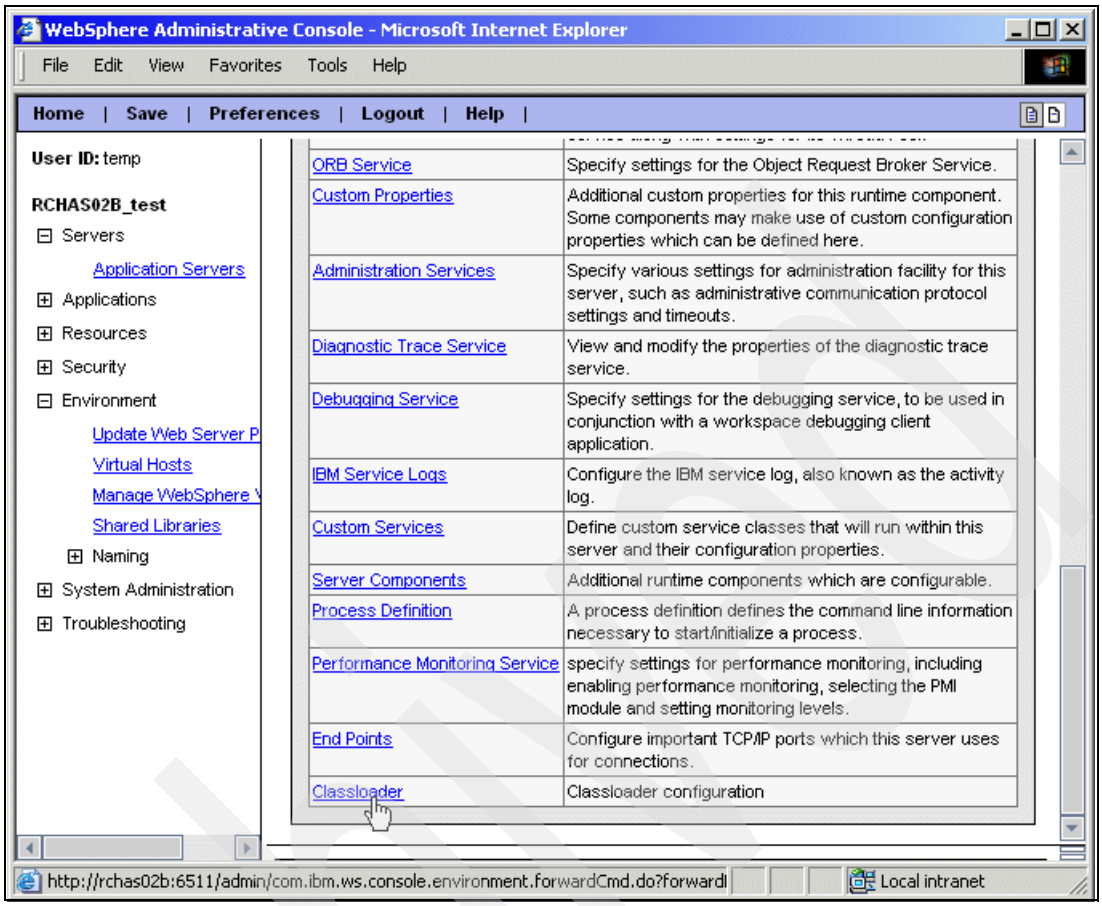


Figure 6-184 Associating a shared library with the application server

5. On the Classloader page, click **New**.
6. Select the **Classloader mode** and click **OK** (see Figure 6-185). The classloader mode defines in what order classloaders are used by WAS.

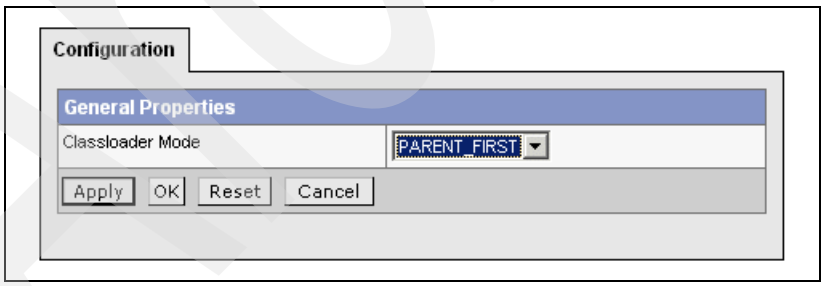


Figure 6-185 Selecting the Classloader mode

7. On the Classloader page, click the classloader that you created. If this is the first classloader that you create for your application server, it is named classloader_1.
8. On the classloader's detail page, click **Libraries**.
9. On the Libraries page, click **Add**.

10. Select the shared library that you want to associate with your application server (see Figure 6-186).

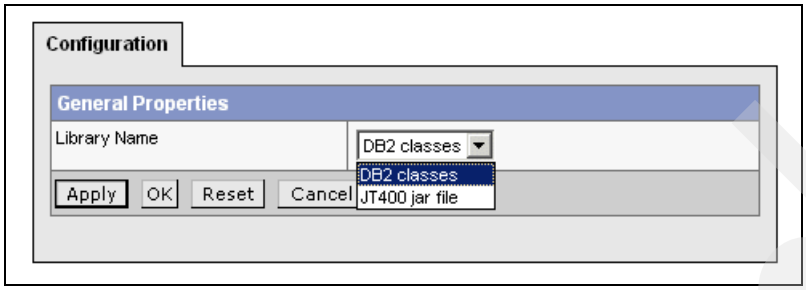


Figure 6-186 Selecting library

11. Click **OK**.
12. **Save** the configuration.
13. Restart the application server.

Removing the association

To remove an association to a shared library, follow these steps:

1. Start the administrative console.
2. In the topology tree, expand **Servers** and click **Application Servers**.
3. On the Application Servers page, click the name of the server from which you want to remove a shared library.
4. On the application server's detail page, click **Classloader** (see Figure 6-184).
5. Click the link for the classloader from which you want to remove a shared library.
6. On the classloader page click **Libraries**.
7. On the Libraries page, select the checkbox for the library that you want to remove.
8. Click **Remove**.
9. **Save** the configuration.
10. Restart the application server.

Associating a shared library with an application

To associate a shared library with an application, follow these steps:

1. Start the administrative console.
2. In the topology tree, expand **Applications** and click **Enterprise Applications**.
3. On the Enterprise Applications page, click the name of the application to which you want to add a shared library.
4. On the application's detail page, click **Libraries**.
5. On the Libraries page, click **Add**.
6. Select the shared library that you want to associate with your application.
7. Click **OK**.
8. **Save** the configuration.
9. Restart the application.

Removing an association

To remove an associations a shared library, follow these steps:

1. Start the administrative console.
2. In the topology tree, expand **Applications** and click **Enterprise Applications**.
3. On the Enterprise Applications page, click the name of the application from which you want to remove a shared library.
4. On the application's detail page, click **Libraries**.
5. On the Libraries page, select the checkbox for the library that you want to remove.
6. Click **Remove**.
7. **Save** the configuration.
8. Restart the application.

6.15 Multi-language support

We can configure individual application servers to run with different globalization settings to support different language environments. The topics in this section discuss how you can configure the application server to run with the desired language environment attributes, and how you can configure the application server to use a specific National Language Version (NLV) when you have multiple WebSphere Application Server NLVs installed.

6.15.1 Configuring the language environment attributes

You can configure the application server to run with the desired language environment attributes, such as the coded character set identifier (CCSID) and country or region identifier. The QEJBSVR user profile settings are the basis for the application server job attributes. The job attributes determine the properties for the JVM environment.

Note: Japanese CCSID 5026 is not supported by WebSphere Application Server. 5035 is the recommended CCSID for this environment. The QEJB and QEJBSVR user profiles default to the system CCSID setting. If the QCCSID system value is set to 5026, the CCSID attribute of the QEJB user profile must be changed to prevent the WebSphere Application Server Monitor and Administration server jobs from trying to start with a CCSID of 5026. The QEJBSVR user profile, and any other user profiles used to run WebSphere Application Server instances, must also be changed.

Also, Arabic CCSID 420 is not supported. Use CCSID 425 instead.

By default, each application server runs under the QEJBSVR user profile. To change the language environment settings, we can either modify the QEJBSVR profile and set the attributes specifically or we can create a new profile which has QJEBSVR as its group profile and which has the appropriate language environment settings. If you choose to create a new profile, you must also register it for use with WebSphere Application Server.

6.15.2 WebSphere Application Server product settings

A second consideration is the national language version (NLV) of the WebSphere Application Server product. The NLV setting only causes the library list to be changed to include the appropriate QSYS*xxxx* library, where *xxxx* is the language feature for which you wish to display messages. The NLV setting does not affect the administrative console or any non-iSeries message logged by the WebSphere Application Server runtime.

Multiple NLVs can be installed at the same time. If the language version that you want to use is the same as the primary language of the system, no additional configuration is needed. If the language version you want to use is a secondary language, configure the application server by setting the *os400.websphere.nlv* property.

6.15.3 Setting the national language version (NLV)

Follow these steps to change NLV:

1. Start the administrative console.
2. Expand **Servers** in the navigation tree.
3. Click **Application Servers**.
4. Click the name of the server for which you wish to change language version.
5. Click the **Configuration** tab, and scroll to **Process Definition** (see Figure 6-187).

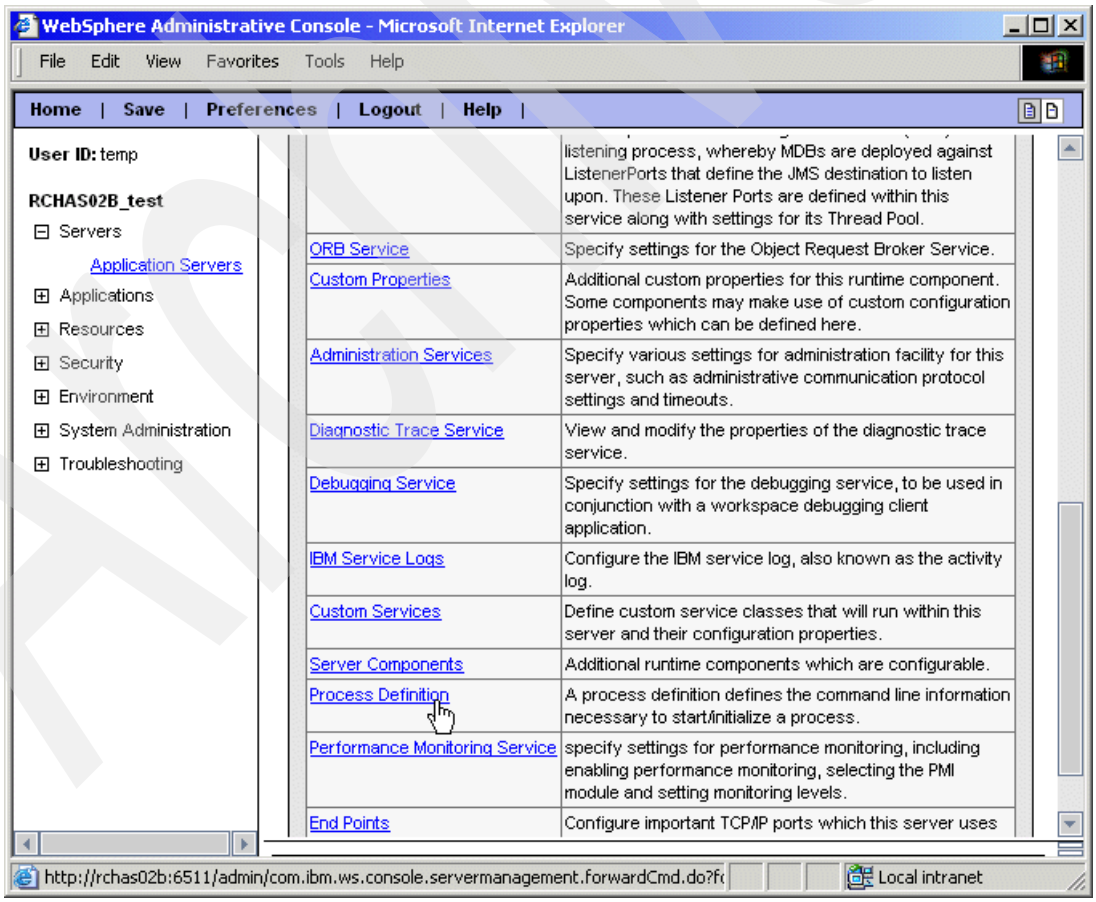


Figure 6-187 Selecting the Process Definition link

6. Click **Java Virtual Machine**.
7. Click **Custom Properties**.
8. Click the **New** button.
9. Specify the following values:

   Name: os400.websphere.nlv

   Value: xxxx

   Here, *xxxx* is the language feature you wish to use to display messages. For example, Japanese language users could specify 2962.
10. Click **OK**.

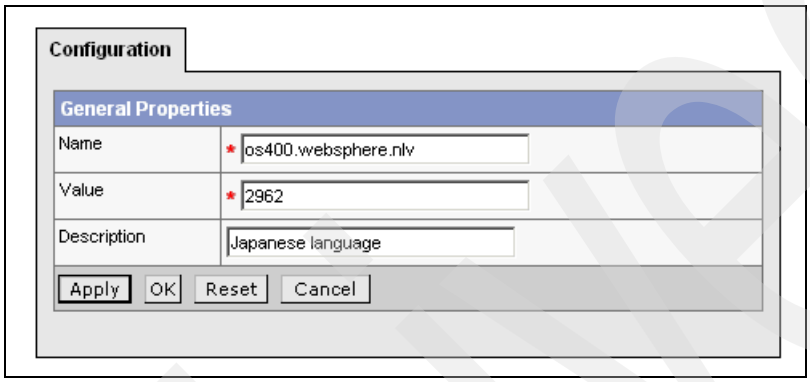


Figure 6-188 The NLV settings

11. Click **Save**.
12. Click **Save** again.
13. After saving the new configuration, restart the application server.

When an application server has the os400.websphere.nlv property set, the corresponding QSYS29*xx* library is added to the beginning of the library list and iSeries messages sent by the application server to the joblog are used accordingly.

7

WebSphere Application Server Network Deployment 5.0: configuration and administration

IBM WebSphere Application Server Network Deployment (WAS-ND) allows one or more IBM WebSphere Application Server nodes to be managed from a single central location. This means that all configuration and management functions for a managed application server in a cell is done from the administrative console, which belongs to the Deployment Manager (DM). DM runs in a WAS-ND instance.

You can use the Deployment Manager to administer application server configurations in a cell. A cell includes configurations for these objects:

- Physical hosts on which application servers are installed
- Application servers and other components that comprise the runtime
- Applications installed on application servers
- Resources providing support to applications, such as JDBC drivers and data sources for data access
- Security

In this chapter we take you through the configuration and administration steps necessary to accomplish these functions.

7.1 Introduction to configuration and administration

The configuration repository of the Deployment Manager is stored in the Integrated File System (IFS) on the iSeries server, where the Deployment Manager resides (the ND environment). This is described in 4.16.2, “Hierarchy of configuration directories” on page 84.

WAS-ND can be installed on any machine in the network to create a Deployment Manager cell. It does not require the IBM WebSphere Application Server to be installed on the same machine, although that is certainly possible. So the WAS-ND and the WebSphere Application Server can be on different machines.

We provide an overview of the major components used in a WAS-ND environment in “Network Deployment topology overview” on page 265.

The Network Deployment product contains a default instance. We describe how to set up this default instance and work with it. Although additional instances can be configured, that is not the scope of this book. You can find a summary describing the necessary steps that have to be done to configure and run the Network Deployment environment in 7.3, “Building a WAS-ND cell” on page 270.

WAS-ND adds clustering and Work Load Management (WLM) capability to WebSphere Application Server Version 5.0. In an application server cluster, workload management distributes client requests among multiple application servers that comprise a cluster. Workload management optimizes the distribution of client processing requests. Incoming work requests are distributed to the application servers, enterprise beans, servlets, and other objects that can most effectively process the requests.

Workload management also provides failover support, improving application availability. In the WAS-ND, workload management is implemented by using clusters of application servers. For more information about workload management; see the online documentation center of WebSphere application server Version 5.0 for iSeries at:

`http://publib.boulder.ibm.com/iseries/v1r1m0/websphere/ic2924/index.htm?info/rzaiz/50/was.htm`

From this site, expand **Administration->High availability->Workload management**.

We also describe the steps to configure clusters (see 7.7, “Cluster support” on page 324).

Note: The ability to run different WebSphere Application Server versions on one server does not allow you to include Version 5 application servers in an existing 3.5 or 4.0 administrative domain, or to include Version 3.5.x or Version 4 application servers in a version 5 cell.

7.2 Network Deployment topology overview

The components of the IBM WebSphere Application Server Network Deployment system management topology are summarized in Figure 7-1.

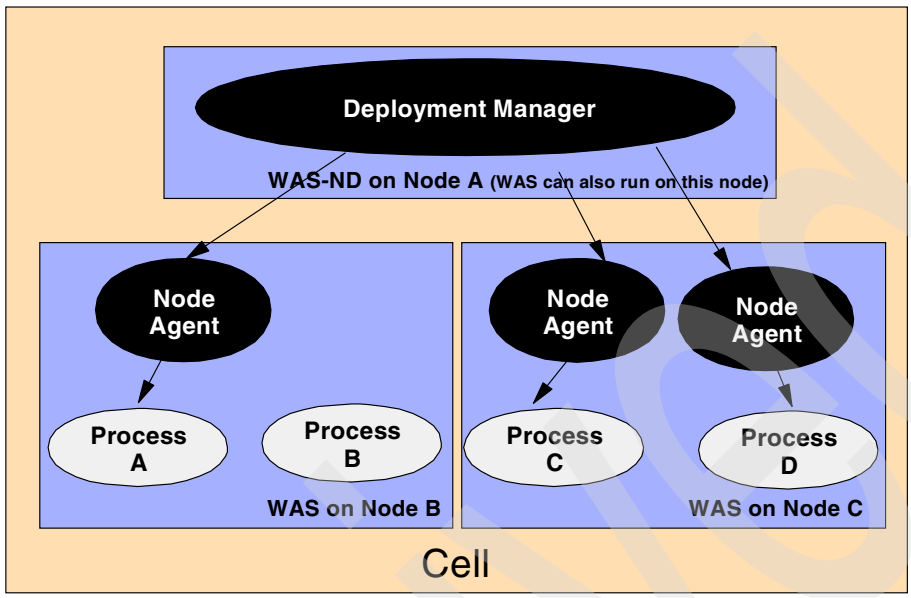


Figure 7-1 Network Deployment system management topology

The major new terms used in a WAS-ND environment include these:

- *Cell* is an aggregation of nodes. A deployment manager controls all nodes that are part of the cell and communicates with them.
- *Node* is a set of managed application servers (instances) on a physical machine in a topology composed of one or more machines. A node contains a IBM WebSphere Application Server installation and is managed by a node agent process. A node cannot span machines, but a single machine can host multiple nodes instances. For every node that you want to be managed by WAS-ND a separate node agent is running.
- *Deployment Manager* (DM) is the process responsible for the management and control of the cell configuration and application data repository. *The Deployment Manager does not handle client requests*, it only maintains the configuration and application information for all nodes (application servers) in the cell.
- *Node Agent* is the process responsible for controlling the managed processes of a node.
- *Managed process* is a single WAS process, running in its own JVM. All operating system processes that are components of the WebSphere product are called managed processes. This means that the processes all participate in the administrative domain.

  Notice also in Figure 7-1 that instance B is not managed by the Deployment Manager of WAS-ND. This means that we can have two or more WAS instances on a node, but not all of them have to be part of a cell.

Figure 7-2 shows the principles of adding a node to an existing cell. The deployment manager will create configuration files for each node that has been added to its cell and assumes responsibility for the configuration of all servers on the node.

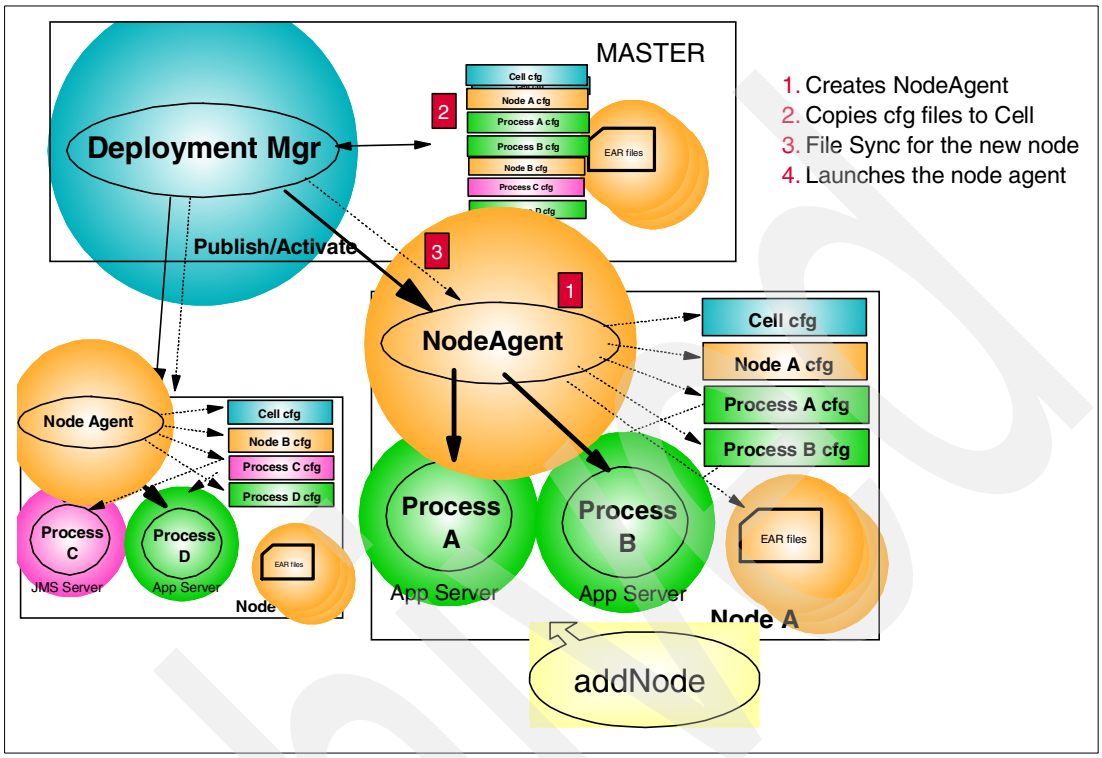


Figure 7-2 Overview add node to a cell

The distributed administration of components is brought about by three tiers (layers) of administration services, as shown in Figure 7-3. Between these tiers, communication is used to distribute configuration and application data updates from the deployment manager to the node agents, and in turn to the server instances.

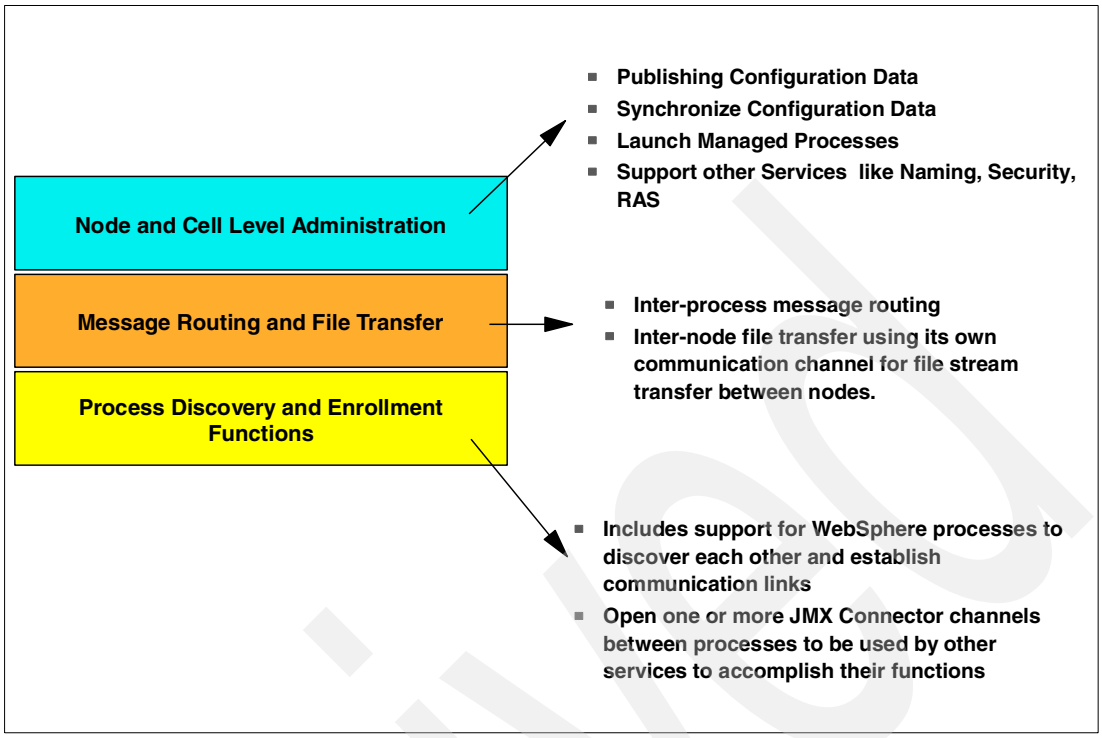


Figure 7-3 Distributed administration services

The routing of administration messages between components makes use of the JMX ObjectName that identifies the target managed resource within the administrative cell (see Figure 7-4). The ObjectName contains all of the information necessary to route a request targeted at the resource, to the appropriate node where the resource is executing.

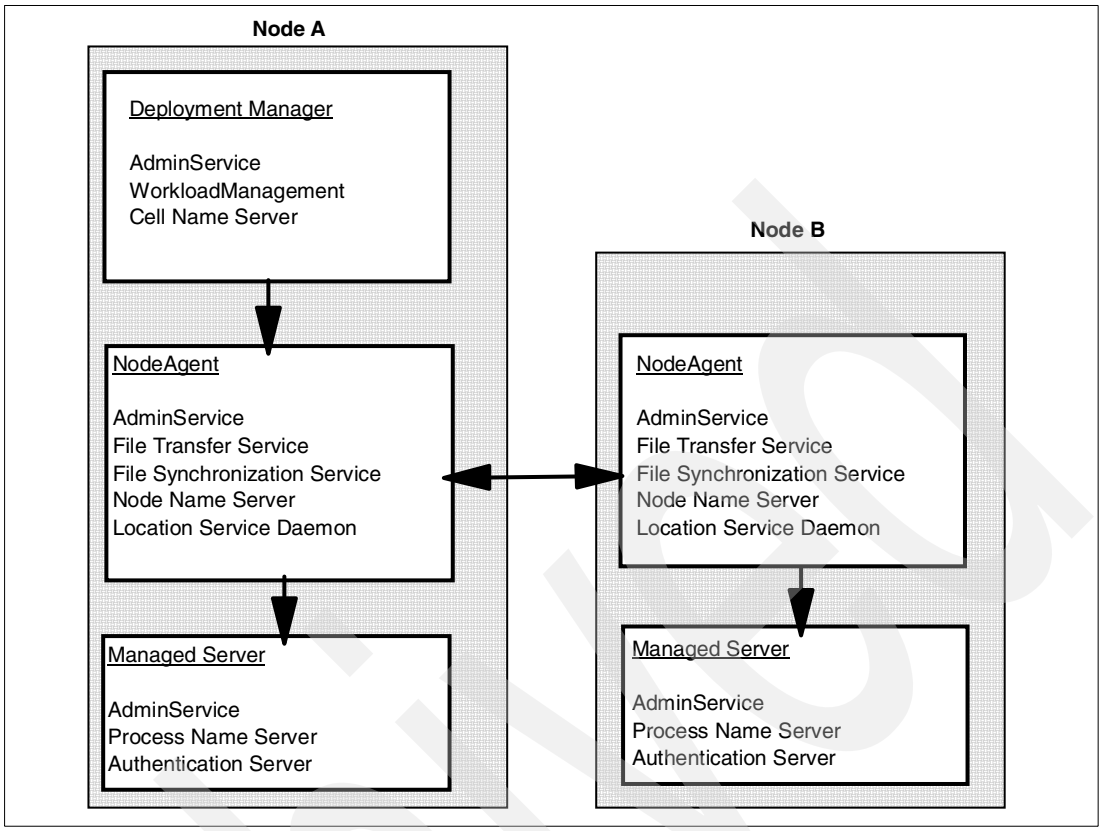


Figure 7-4 Distributed administration communication flow

Modifying configuration settings in an ND environment can be done by the following administration client programs:

- QShell scripts
- Web-based administration (administrative console)
- WebSphere scripting (wsadmin)

When configuration changes are made, the Deployment Manager updates the Node Agents in the domain at regular intervals; the default is 1 Minute. You can change this default value; see 7.4.3, “Synchronizing configuration changes” on page 304.

The Deployment Manager communicates with Node Agents by way of a configurable Discovery Address — a port and protocol dedicated to letting WebSphere processes find each other. All ports and services the Deployment Manager uses for communication is stored in the serverindex.xml file. For an example of the contents of this file for the Deployment Manager default instance; see Figure 7-13 on page 278.

The Node Agent is itself an application server process, running a File Transfer enterprise application. The Node Agent runs in the subsystem QEJBAS5, so on the node where the application server resides. This application exchanges information with the Deployment Manager. It creates a local copy of all the information that the Deployment Manager sends it.

Each node (WAS instance) has its own repository with local configuration files. Changes made to a local node are temporary if the node belongs to a cell. While in effect, local changes override cell configuration. Changes at the cell level to a node and server files are permanent. Synchronization occurs at designated intervals or events, such as when the server starts. See also the information about the communication between the Deployment Manager and the Node Agents described above.

Note: The cell repository (managed by the Deployment Manager) is considered the master repository. Configuration changes made to node repositories are not propagated up to the cell.

In this chapter we describe the QShell scripts and some of the functions of the administrative console that are important in the ND environment.

7.2.1 QShell scripts and administration functions used in this chapter

In general, you can use the administrative console in a Network Deployment environment for operational administration of the ND cell. Also, IBM WebSphere Application Server provides several script commands that can be used via the QShell interface of the iSeries server for operational administration of the WAS servers in an ND cell and the Network Deployment Manager.

Table 7-1 shows only the script commands we used in this chapter. In the column *Pointer* we refer to the sections that describe the scripts. Also, in this column, we mention when we use the administrative console to perform the same task.

Instead of using the QShell scripts, you can also use the wsadmin tool (see Chapter 9, “The wsadmin tool” on page 385).

Table 7-1 used QShell script commands

Command	Usage	Pointer	Syntax reference
Script commands and administrative console functions used for server management			
dspwasinst	Display WAS instance	Figure 7-5 on page 272 Figure 7-6 on page 273 Figure 7-8 on page 274 Figure 7-45 on page 309	
chgwassvr	Change WAS server	Figure 7-7 on page 273, and “Changing a server in an ND environment via a script” on page 289	A.7.4, “Syntax and parameters for chgwassvr script” on page 534
		7.5, “Changing a server via the administrative console” on page 315	
startServer	Start WAS server		A.7.2, “Syntax and parameters for startServer script” on page 532
stopServer	Stop WAS server		A.7.3, “Syntax and parameters for stopServer script” on page 533
		7.4.8, “Starting/stopping an application server via the admin console” on page 314	
	Start/stop JMS server	7.5.2, “Starting and stopping a JMS server in the ND environment” on page 319	

Command	Usage	Pointer	Syntax reference
	Change JMS server	7.5.3, “Changing the JMS server configuration in the ND environment” on page 321	
script commands and administrative console functions used to administer nodes in an ND environment			
addNode	Add node to cell	7.3.2, “Adding a node to the network deployment instance” on page 280	A.7.8, “The syntax of the addNode script” on page 541
		7.4.5, “Adding a node via the administrative console” on page 307	
startNode	Start Node Agent	“Starting a node agent via startNode script” on page 291	A.7.9, “The syntax of the startNode script” on page 543
stopNode	stop Node Agent	“Stopping a node agent via stopNode script” on page 295	A.7.10, “The syntax of the stopNode script” on page 544
	Start/stop Node Agents	7.4.6, “Managing node agents via the ND admin console” on page 310	
syncNode	Synchronize node with cell	“Using the syncNode script” on page 290	
		“Synchronizing the configuration via the ND administrative console” on page 306	
removeNode	Remove node	“The removeNode script” on page 296	
		“Remove a node via the administrative console” on page 296	
cleanupNode		“Cleanup node” on page 297	
Script commands and administrative console functions used to manage the Deployment Manager			
startManager	Start the deployment manager for an ND instance	“The startManager script” on page 279	A.7.6, “Syntax and parameters of the startManager script” on page 538
stopManager	Stop the deployment manager for an ND instance	“The stopManager script” on page 279	A.7.7, “Syntax and parameters for the stopManager script” on page 539
Other script commands			
GenPluginCfg	Regenerate the plugin-cfg.xml file	“The GenPluginCfg script” on page 302	A.7.11, “The syntax of the GenPluginCfg script” on page 546

7.3 Building a WAS-ND cell

In this section we describe what basic administration steps comprise the task of creating a WAS-ND cell. We structure this section following the WAS-ND topology:

- Administrative steps to manage Deployment Manager (DM)
- Administrative steps to add a node to the cell
- Administrative steps to manage a node in the cell

7.3.1 Administering the Deployment Manager node

First we need to understand what tasks are involved in managing a DM node. This section will provide this information.

Before you start the ND environment

Before you start the WAS Network Deployment environment for the first time, you should examine your installation to make sure you can prevent port conflicts.

As the WebSphere Application Server, the WebSphere Application Server Network Deployment is shipped with a default instance with a default Deployment Manager that will be started when you start the iSeries subsystem for the ND. See “Starting Deployment Manager” on page 274.

This default WAS-ND instance (Deployment Manager) uses the same port numbers for the administrative console port (default is 9090) and administrative console SSL-enabled port (default is 9043) as the WebSphere Application server for the default instance.

When you plan to run WebSphere Application Server Version 5.0 and WebSphere Application Server Network Deployment Version 5.0 on the same iSeries server, you may have to change the administrative ports for one of the default instances. This is the case if you are not adding the default instance of the Base product to a cell.

In our scenario we have first installed the WAS software on our iSeries system, then we installed the WAS ND software.

First we use the `dspwasinst` script to display the port values for the WAS default instance. To do this:

1. Start the QShell from the OS/400 command line:

   ```
   STRQSH
   ```

2. Change to the following directory:

   ```
   cd /qibm/ProdData/WebAS5/Base/bin
   ```

3. Run the `dspwasinst` script:

   ```
   dspwasinst -instance default
   ```

Figure 7-5 shows the result for the WAS default instance.

```
dspwasinst -instance default
ADCP0005I: Using cell RCHAS02B, node RCHAS02B and server *ALL.
Display WAS instance:
  Instance name: default
  Instance type: Base Application Server
  Cell: RCHAS02B
  Node: RCHAS02B

  Information for server: server1
    Installed applications:
      DefaultApplication
      ivtApp
      adminconsole

     Ports in use:
          9090   Administrative console port
          9043   Administrative console SSL-enabled port
          2809   Name service port
          8880   Soap port
          9080   Internal HTTP port
          7873   Data replication service client port
          5557   Internal Java Message Service server port
          5558   Queued Java Message Service server port
          5559   Direct Java Message Service server port
          9501   SAS SSL server authentication listener port
          9503   CSIV2 server authentication listener port
          9502   CSIV2 mutual authentication listener port
$
```

Figure 7-5 Default WAS instance from /qibm/proddata/webas5/base/bin

Second, we use the `dspwasinst` script to display the port values for the ND default instance. To do so:

1. Start the QShell from the OS/400 command line:

   ```
   STRQSH
   ```

2. Change to the following directory:

   ```
   cd /qibm/ProdData/WebAS5/ND/bin
   ```

3. Run the `dspwasinst` script:

   ```
   dspwasinst -instance default
   ```

Figure 7-6 shows the result for the ND default instance.

Important: Note that we ran the `dspwasinst` script from two different directories: one time from the WAS (Base) directory, and a second time from the WAS-ND directory.

Compare the ports in use for the WAS and ND instances; you will see that the ports for the administrative console are the same. We decided to change the port numbers for the ND default instance.

```
dspwasinst -instance default
ADCP0005I: Using cell RCHAS02BNetwork, node RCHAS02BManager and server *ALL.
Display WAS instance:
  Instance name: default
  Instance type: Network Deployment
  Cell: RCHAS02BNetwork
  Node: RCHAS02BManager

  Managed nodes:
    None

  Information for server: dmgr
    Installed applications:
      filetransfer
      adminconsole

    Ports in use:
         9090   Administrative console port
         9043   Administrative console SSL-enabled port
         9809   Name service port
         8879   Soap port
         7989   Data replication service client port
         9401   SAS SSL server authentication listener port
         9403   CSIV2 server authentication listener port
         9402   CSIV2 mutual authentication listener port
         9100   ORB listener port
         7277   Cell discovery port
$
```

Figure 7-6 Default ND instance from /qibm/proddata/webas5/nd/bin

To change the adminconsole ports for the ND default instance, use the `chgwassvr` script. We changed the administrative console port to 9095 and the administrative console SSL-enabled port to 9045.

1. To do so, start the QShell from an OS/400 command line and press Enter.

   ```
   STRQSH
   ```

2. On the QShell command line, use the **cd** command to change to the WAS ND directory:

   ```
   cd /QIBM/ProdData/WebAS5/ND/bin
   ```

3. Use the `chgwassvr` script (for more information see 6.5.7, “Changing a WAS application server via the script” on page 128).

   ```
   chgwassvr -instance default -server dmgr -admin 9095 -adminssl 9045
   ```

   Figure 7-7 shows the message for the `chgwassvr` script used to change the ND default instance.

```
chgwassvr -instance default -server dmgr -admin 9095 -adminssl 9045
ADCP0005I: Using cell RCHAS02BNetwork, node RCHAS02BManager and server dmgr.
ADCP0001I: Ports changed successfully.
ADCP0002I: The following ports were changed to the specified value:
             Administrative console port:   9,095
             Administrative console SSL-enabled port:   9,045
$
```

Figure 7-7 change admin and adminssl port via chgwasinst from ND directory

4. Run the dspwasinst script:

   ```
   dspwasinst -instance default
   ```

   Figure 7-8 shows the value for the changed ND default instance.

```
dspwasinst -instance default
ADCP0005I: Using cell RCHAS02BNetwork, node RCHAS02BManager and server *ALL.
Display WAS instance:
  Instance name: default
  Instance type: Network Deployment
  Cell: RCHAS02BNetwork
  Node: RCHAS02BManager

  Managed nodes:
    None

  Information for server: dmgr

   Installed applications:
     filetransfer
     adminconsole

   Ports in use:
        9095   Administrative console port
        9045   Administrative console SSL-enabled port
        9809   Name service port
        8879   Soap port
        7989   Data replication service client port
        9401   SAS SSL server authentication listener port
        9403   CSIV2 server authentication listener port
        9402   CSIV2 mutual authentication listener port
        9100   ORB listener port
        7277   Cell discovery port
```

Figure 7-8 Changed ND default instance

Starting Deployment Manager

The Deployment Manager of a WAS-ND instance runs in a job in the QEJBASND5 subsystem. By default, the subsystem QEJBASND5 has an autostart job entry to start the default instance with the Deployment Manager job with the name DMGR.

You need a user profile that has *JOBCTL authority to the user profile QEJBSVR to start the QEJBASND5 subsystem.

To start the QEJBASND5 subsystem, enter this command on an OS/400 command line and press Enter:

```
STRSBS QEJBAS5/QEJBASND5
```

The default DM instance starts too. If you need to start a non-default DM instance; see “The startManager script” on page 279.

Verifying that the Deployment Manager has started

Before you start the administrative console or add a node to the Network Deployment cell, you should verify that the Network Deployment environment has started successfully.

When the Network Deployment environment is ready for use, a message is written to the job log of the Deployment Manager job, indicating that the Network Deployment environment is ready. When you find the message “WebSphere application server *server_name* ready”, where *server-name* is the name of the Deployment Manager, the Network Deployment environment has successfully started.

To determine if the Deployment Manager is ready, perform these steps from an OS/400 command line:

1. Run the Work with Active Jobs (WRKACTJOB) command, specifying the QEJBASND5 subsystem on the subsystem (SBS) parameter and press Enter:

   ```
   WRKACTJOB SBS(QEJBASND5)
   ```

2. Find the server job for your Network Deployment instance. For the default instance, the server job for the Deployment Manager is named DMGR, as shown in Figure 7-9.

```
                          Work with Active Jobs                    RCHAS02B
                                                        12/17/02  19:11:04
CPU %:     .0     Elapsed time:   00:00:00     Active jobs:   251

Type options, press Enter.
  2=Change   3=Hold   4=End   5=Work with   6=Release   7=Display message
  8=Work with spooled files   13=Disconnect ...

Opt  Subsystem/Job  User        Type  CPU %  Function        Status
     QEJBASND5      QSYS        SBS      .0                 DEQW
       DMGR         QEJBSVR     ASJ      .0  PGM-QEJBSTRSVR   JVAW




                                                                    Bottom
Parameters or command
===>
F3=Exit   F5=Refresh       F7=Find      F10=Restart statistics
F11=Display elapsed data   F12=Cancel   F23=More options   F24=More keys
```

Figure 7-9 Deployment Manager job

3. Specify option **5** (Work with Job) on the option line next to the job, and press Enter.
4. On the command line of the Work with Job display, specify option **10** (Display job log, if active), and press Enter. You will see a screen like the one shown in Figure 7-10.
5. Press F10 (Display detailed message).

```
Display Job Log
                                                          System:   RCHAS02B
Job . . :   DMGR          User . . :   QEJBSVR       Number . . . :   023514

 >> QSYS/CALL PGM(QEJBAS5/QEJBSTRSVR) PARM('-instance' '/QIBM/UserData/WebAS5/
    ND/default' '-server' 'dmgr')









                                                                        Bottom
Press Enter to continue.

F3=Exit   F5=Refresh   F10=Display detailed messages   F12=Cancel
F16=Job menu           F24=More keys
```

Figure 7-10 Deployment Manager job

6. Look for this message:

   ```
   WebSphere application server application_server ready.
   ```

   Here, *application_server* is the name of your Deployment Manager. For the default Network Deployment instance, the deployment manager is named dmgr; see Figure 7-11 on page 276.

   If the message is not displayed, press F5 to refresh the job log messages until the message is displayed. If the message is still not displayed, refer to Chapter 11, “Troubleshooting” on page 431.

```
                              Display All Messages
                                                          System:   RCHAS02B
Job . . :   DMGR          User . . :   QEJBSVR       Number . . . :   023514

 >> QSYS/CALL PGM(QEJBAS5/QEJBSTRSVR) PARM('-instance' '/QIBM/UserData/WebAS5/
    ND/default' '-server' 'dmgr')
    Server starting with user profile QEJBSVR and JDK 1.3.1.
    WebSphere application server dmgr ready.


                                                                        Bottom
Press Enter to continue.

F3=Exit   F5=Refresh   F12=Cancel   F17=Top   F18=Bottom
```

Figure 7-11 Port number used to communicate with the administrative console

7. To display the port number on which the administrative console of the Deployment Manager is listening, position the cursor on the “ready” line and press F1.

   The Additional Message Information display shows the port number, as you can see in Figure 7-12.

Note: Normally the default port for the administrative console is 9090, but we have changed the port to 9095 prior to starting DM; see “Building a WAS-ND cell” on page 270.

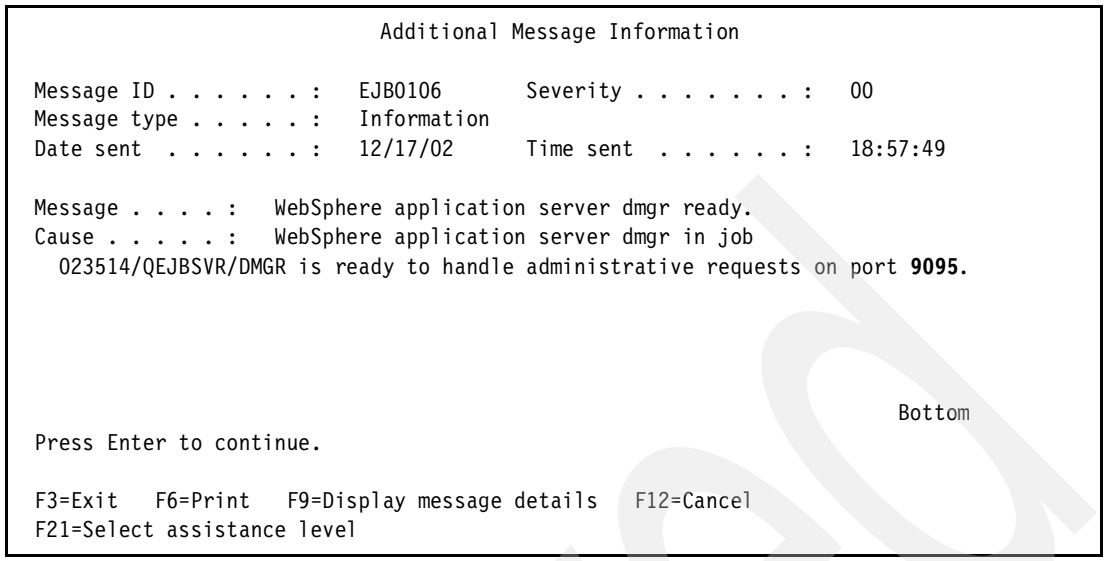


Figure 7-12 TCP/IP port for the administrative console in ND

8. Press F3 to exit.

Starting the ND subsystem automatically at iSeries system start up

You can configure your iSeries in such a way that the QEJBASND5 subsystem automatically starts at iSeries system startup (IPL). To do so, add an autostart job entry to the QSYSWRK subsystem:

- Enter this command on an iSeries command line and press Enter:

  ```
  ADDAJE SBSD(QSYSWRK) JOB(QEJBSTRTND) JOBD(QEJBAS5/QEJBSTRTND)
  ```

  You will get this message:

  ```
  Change effective next time subsystem starts.
  ```

Deployment Manager's *serverindex.xml* file

All configuration details of DM are stored in the XML files in IFS. One of the key configuration files is *serverindex.xml*.

Figure 7-13 shows an example of the serverindex.xml file that is used for the default instance of the Deployment Manager. This file resides in the following directory, where *system_name* is the TCP/IP name of your iSeries server:

```
/QIBM/UserData/WebAS5/ND/default/config/cells/system_nameNetwork/nodes/system_nameManager
```

This file contains the ports for the different services that are assigned to the default WAS-ND instance. Notice that the ports used for the administrative console are not in this file, they are stored in the following directories (replace <*system_name*> with the host name of your iSeries system):

- The *virtualhosts.xml* file is in the directory:

  ```
  /QIBM/UserData/WebAS5/ND/default/config/cells/system_nameNetwork
  ```

- The *server.xml* file is in the directory:

  ```
  /QIBM/UserData/WebAS5/ND/default/config/cells/<system_name>Network/nodes/
  <system_name>Manager/servers/dmgr
  ```

```
<?xml version="1.0" encoding="UTF-8"?>
<xmi:XMI xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
xmlns:serverindex="http://www.ibm.com/websphere/appserver/schemas/5.0/serverindex.xmi"
xmlns:ipc="http://www.ibm.com/websphere/appserver/schemas/5.0/ipc.xmi">
  <xmi:Documentation>
    <contact>WebSphere Application Server v5.0 Default Configuration Files v1.25
8/15/02</contact>
  </xmi:Documentation>
  <serverindex:ServerIndex xmi:id="ServerIndex_1" hostName="RCHAS02B"
endPointRefs="NamedEndPoint_1 NamedEndPoint_2">
    <serverEntries xmi:id="ServerEntry_1" serverDisplayName="dmgr" serverName="dmgr"
serverType="DEPLOYMENT_MANAGER">

<deployedApplications>filetransfer.ear/deployments/filetransfer</deployedApplications>

<deployedApplications>adminconsole.ear/deployments/adminconsole</deployedApplications>
      <specialEndpoints xmi:id="NamedEndPoint_1" endPointName="CELL_DISCOVERY_ADDRESS">
        <endPoint xmi:id="EndPoint_1" host="RCHAS02B" port="7277"/>
      </specialEndpoints>
      <specialEndpoints xmi:id="NamedEndPoint_2" endPointName="BOOTSTRAP_ADDRESS">
        <endPoint xmi:id="EndPoint_2" host="RCHAS02B" port="9809"/>
      </specialEndpoints>
      <specialEndpoints xmi:id="NamedEndPoint_3" endPointName="DRS_CLIENT_ADDRESS">
        <endPoint xmi:id="EndPoint_3" host="RCHAS02B" port="7989"/>
      </specialEndpoints>
      <specialEndpoints xmi:id="NamedEndPoint_4" endPointName="SOAP_CONNECTOR_ADDRESS">
        <endPoint xmi:id="EndPoint_4" host="RCHAS02B" port="8879"/>
      </specialEndpoints>
      <specialEndpoints xmi:id="NamedEndPoint_5" endPointName="ORB_LISTENER_ADDRESS">
        <endPoint xmi:id="EndPoint_5" host="RCHAS02B" port="9100"/>
      </specialEndpoints>
      <specialEndpoints xmi:id="NamedEndPoint_6"
endPointName="SAS_SSL_SERVERAUTH_LISTENER_ADDRESS">
        <endPoint xmi:id="EndPoint_6" host="RCHAS02B" port="9401"/>
      </specialEndpoints>
      <specialEndpoints xmi:id="NamedEndPoint_7"
endPointName="CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS">
        <endPoint xmi:id="EndPoint_7" host="RCHAS02B" port="9402"/>
      </specialEndpoints>
      <specialEndpoints xmi:id="NamedEndPoint_8"
endPointName="CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS">
        <endPoint xmi:id="EndPoint_8" host="RCHAS02B" port="9403"/>
      </specialEndpoints>
    </serverEntries>
  </serverindex:ServerIndex>
</xmi:XMI>
```

Figure 7-13 Serverindex.xml file for the default deployment manager instance

Scripts for managing the Deployment Manager

You can start and stop the Deployment Manager for an ND instance via the appropriate scripts, `startManager` and `stopManager`, from a QShell command line interface:

1. Start the QShell from an OS/400 command line and press Enter:

   ```
   STRQSH
   ```

2. On the QShell command line, use the **cd** command to change to the WAS ND directory:

   ```
   cd /QIBM/ProdData/WebAS5/ND/bin
   ```

3. Type the `startManager` or the `stopManager` command following the syntax and using the needed parameters; see “The startManager script” on page 279 and “The stopManager script” on page 279.

The startManager script

The `startManager` script starts the deployment manager process for a Network Deployment instance and is available with WebSphere Application Server Network Deployment only.

To run this script, your user profile must have *ALLOBJ authority.

Examples:

1. `startManager`

   This example starts the default deployment manager (dmgr job on iSeries) for the default instance.

2. `startManager -instance myinst`

   If you omit the name of the server, it defaults to the name of the instance (one exception: for the `default` instance, the name of the server defaults to `dmgr`). This example starts the `myinst` deployment manager in the `myinst` instance.

3. `startManager test -instance devinst -nowait`

   This example starts the `test` deployment manager in the `devinst` instance and returns control immediately to the user.

For the detailed syntax; see A.7.6, “Syntax and parameters of the startManager script” on page 538

The stopManager script

The `stopManager` script stops the deployment manager process for a Network Deployment instance. The `stopManager` script is available with WebSphere Application Server Network Deployment only.

To run this script, your user profile must have *ALLOBJ authority.

Examples:

1. `stopManager dmgr`

   This stops the default Deployment Manager of the default instance; see Figure 7-14.

```
stopManager dmgr
ADMU0116I: Tool information is being logged in file
           /QIBM/UserData/WebAS5/ND/default/logs/dmgr/stopServer.log
ADMU3100I: Reading configuration for server: dmgr
ADMU3201I: Server stop request issued. Waiting for stop status.
ADMU4000I: Server dmgr stop completed.
$
```

Figure 7-14 stop Deployment Manager

2. `stopManager myserver -instance myinst -port 10380 -conntype SOAP`

   This example uses the SOAP port (10380) to stop the myserver deployment manager in the myinst instance.

For the detailed syntax, see A.7.7, "Syntax and parameters for the stopManager script" on page 539

7.3.2 Adding a node to the network deployment instance

To add a node to a cell, you can use the `addNode` script or the administrative console from the ND environment. We recommend that you always use the `addNode` script, because the administrative console doesn't provide a way to specify a user-defined port numbers, so the default ports are assigned to the federated Node Agent and JMS server.

To see an example of how to use the administrative console to add a node, and also why you should not use this approach, see 7.4.5, "Adding a node via the administrative console" on page 307.

The `addNode` script is available in the WebSphere Application Server product only. Use this script to add WebSphere Application Server nodes (server) to the cell which your Network Deployment instance is managing. You run this script from the node (system) where your WAS instances is running.

Repeat the steps below for each node (or application server) you wish to add to the ND domain.

When you add a server (node) to a deployment manager cell, WebSphere Application Server creates a Node Agent, named *nodeagent*.

To run the `addNode` script, your user profile must have *ALLOBJ authority. Perform the following steps to add a node:

1. The Deployment Manager must be started and actively listening on the port specified for the SOAP_CONNECTOR_ADDRESS port (default 8879) before you can add a node to the Deployment Manager cell. Refer to "Verifying that the Deployment Manager has started" on page 275.
2. On the OS/400 command line of the system which contains the node that you want to add, enter the STRQSH (Start Qshell) command and press Enter.
3. On the Qshell command line, use the **cd** command to change to the directory that contains the script:

   ```
   cd /QIBM/ProdData/WebAS5/Base/bin
   ```

4. Invoke the addNode script specifying the host name and Simple Object Access Protocol (SOAP) port for the Deployment Manager:

   ```
   addNode hostName soapPort
   ```

   Note: The hostName and soapPort parameters are positional and must be the first two parameters. Additional parameters such as -includeapps must be specified after the soapPort value.

   a. **Example 1:** In the first example we add a default instance to the default cell. ND runs on the RCHAS02B system. Port 8879 is the default SOAP port for the default instance of DM:

      ```
      addNode RCHAS02B 8879
      ```

      The output of this command shown in the Qshell display. For our example A, you can see this message in Figure 7-15 on page 282, Figure 7-16 on page 283 and Figure 7-17 on page 284.

      Observing the output, we can identify several key steps in federating a node to a cell:

      - The application server server1 is stopped (the server was active when we called the addNode script).
      - The Node Agent is created and started.
      - Synchronizing configuration between node and cell has been done.
      - A Queue Manager for node RCHAS02B has been created, because our *server 1* was enabled for embedded JMS.
      - Because -includeapps was not specified in our addNode script, the installed applications (in our case, the default applications) are not uploaded to the cell.
      - The server plugin configuration file for all servers in cell (RCHAS02BNetwork) is updated.
      - The ports that the addNode script assigned to the Node Agent and the JMS server are the default ports.
      - The addnode.log file is created in the log directory under the root directory of the instance (for this example it is in /QIBM/UserData/WebAS5/Base/default/logs). This file contains the system output of the process of adding a new node.
      - The serverindex.xml configuration file for the federated node server1 is also created; see Figure 7-18 on page 285 and Figure 7-19 on page 286.

   b. **Example 2:** We add our second instance on RCHAS02B to the Network Deployment cell. The instance name is webface:

      ```
      addNode RCHAS02B 8879 -instance webface -startingport 20300 -includeapps
      ```

      The startingport parameter allows you to specify the first port number out of 12 consecutive ports used by the Node Agent. See the messages that are shown in the QShell in Figure 7-20 on page 287 and Figure 7-21 on page 288. They are quite similar to the messages from the previous example.

      By default, any applications installed on the WebSphere Application Server are removed from the server when it is added to a cell. You can specify the -includeapps parameter when invoking addNode. By using this parameter, we indicate that all applications from the federated node have to be uploaded to the DM system.

Tip: It is recommended to use the -startingport parameter with the addNode script to avoid port conflicts.

```
addNode RCHAS02B 8879
CPI1462:  Change effective next time subsystem starts.
CPC1604:  Active subsystem description QEJBAS5 in QEJBAS5 changed.
CPF1010:  Subsystem name QEJBAS5 active.
ADMU0116I: Tool information is being logged in file
           /QIBM/UserData/WebAS5/Base/default/logs/addNode.log
ADMU0001I: Begin federation of node RCHAS02B with Deployment Manager at
           RCHAS02B:8879.
ADMU0009I: Successfully connected to Deployment Manager Server: RCHAS02B:8879
ADMU0505I: Servers found in configuration:
ADMU0506I: Server name: server1
ADMU2010I: Stopping all server processes for node RCHAS02B
ADMU0510I: Server server1 is now STOPPED
ADMU0024I: Deleting the old backup directory.
ADMU0015I: Backing up the original cell repository.
ADMU0012I: Creating Node Agent configuration for node: RCHAS02B
ADMU0014I: Adding node RCHAS02B configuration to cell: RCHAS02BNetwork
ADMU0016I: Synchronizing configuration between node and cell.
ADMU0018I: Launching Node Agent process for node: RCHAS02B
ADMU0020I: Reading configuration for Node Agent process: nodeagent
ADMU0022I: Node Agent launched. Waiting for initialization status.
ADMU0030I: Node Agent initialization completed successfully. Process id is:
           023533/QEJBSVR/NODEAGENT
ADMU0523I: Creating Queue Manager for node RCHAS02B on server jmsserver
ADMU0525I: Details of Queue Manager creation may be seen in the file:
           createMQ.RCHAS02B_jmsserver.log
ADMU9990I:
ADMU0300I: Congratulations! Your node RCHAS02B has been successfully
            incorporated into the RCHAS02BNetwork cell.
 ADMU9990I:
 ADMU0306I: Be aware:
 ADMU0302I: Any cell-level documents from the standalone RCHAS02B configuration
            have not been migrated to the new cell.
 ADMU0307I: You might want to:
 ADMU0303I: Update the configuration on the RCHAS02BNetwork Deployment Manager
            with values from the old cell-level documents.
 ADMU9990I:
 ADMU0306I: Be aware:
 ADMU0304I: Because -includeapps was not specified, applications installed on
            the standalone node were not installed on the new cell.
 ADMU0307I: You might want to:
 ADMU0305I: Install applications onto the RCHAS02BNetwork cell using wsadmin
            $AdminApp or the Administrative Console.
 ADMU9990I:
 ADMU0003I: Node RCHAS02B has been successfully federated.
IBMWebSphereApplicationServer,Release5.0 WebSpherePluginConfigurationGenerator
Copyright IBM Corp., 1997-2002

PLGC0013I: Generating server plugin configuration file for all of the servers in cell
RCHAS02BNetwork.

PLGC0022W: No definition found for server dmgr.  Server will be ignored.

PLGC0020E: No valid server definitions found for server cluster
dmgr_RCHAS02BManager_Cluster. It will be ignored.
```

Figure 7-15 Messages to addNode server default, example A - part 1 of 3

```
 PLGC0005I: Plugin Configuration file =
/QIBM/UserData/WebAS5/Base/default/config/cells/plugin-cfg.xml
 ADCP0005I: Using cell RCHAS02BNetwork, node RCHAS02B and server *ALL.
 Display WAS instance:
   Instance name: default
    Instance type: Federated Base Application Server

      Cell: RCHAS02BNetwork
      Node: RCHAS02B

      Deployment manager for node: dmgr
        Deployment manager host: RCHAS02B
        Deployment manager SOAP port: 8879
        Deployment manager RMI port: 9809

      Information for server: server1
        Installed applications:
          None

        Ports in use:
             9810   Name service port
             8880   Soap port
             9080   Internal HTTP port
             7873   Data replication service client port


             9501   SAS SSL server authentication listener port
             9503   CSIV2 server authentication listener port
             9502   CSIV2 mutual authentication listener port

      Information for server: nodeagent
        Installed applications:
          None

        Ports in use:
             2809   Name service port
             8878   Soap port
             7888   Data replication service client port
             9901   SAS SSL server authentication listener port
             9201   CSIV2 server authentication listener port
             9101   CSIV2 mutual authentication listener port
             9900   ORB listener port
             7272   Node discovery port
             5000   Node multicast discovery port
```

Figure 7-16 Messages to addNode server default, example A - part 2 of 3

```
Information for server: jmsserver
     Installed applications:
       None

     Ports in use:
          2810   Name service port
          8876   Soap port
          5557   Internal Java Message Service server port
          5558   Queued Java Message Service server port
          5559   Direct Java Message Service server port

 CPC1612:  Job description QEJBJOBD in library QEJBAS5 changed.
 CPI1462:  Change effective next time subsystem starts.
 CPC1604:  Active subsystem description QEJBAS5 in QEJBAS5 changed.
 $
```

Figure 7-17 Messages to addNode server default, example A - part 3 of 3

Figure 7-18 on page 285 and Figure 7-19 on page 286 show the example of the serverindex.xml file for our default instance after the node has been federated to the cell. It is located in the configuration directory of the default instance:

```
/QIBM/UserData/WebAS5/Base/default/config/cells/cellName/nodes/nodeName
```

Here, `cellName` is the name of your cell and `nodeName` is the name of your node.

Notice that the ports used for the administrative console are not in this file, they are stored in the *virtualhosts.xml* file located on the system where you run DM.

```
<?xml version="1.0" encoding="UTF-8"?>
<xmi:XMI xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
xmlns:serverindex="http://www.ibm.com/websphere/appserver/schemas/5.0/
 <xmi:Documentation>
    <contact>WebSphere Application Server v5.0 Default Configuration Files v1.12
8/15/02</contact>
 </xmi:Documentation>
  <serverindex:ServerIndex xmi:id="ServerIndex_1" hostName="RCHASO2B"
endPointRefs="NamedEndPoint_4 NamedEndPoint_5 NamedEndPoint_9
    <serverEntries xmi:id="ServerEntry_1" serverDisplayName="server1"
serverName="server1" serverType="APPLICATION_SERVER">
  <specialEndpoints xmi:id="NamedEndPoint_1" endPointName="BOOTSTRAP_ADDRESS">
    <endPoint xmi:id="EndPoint_1" host="RCHASO2B" port="9810"/>
    </specialEndpoints>
 <specialEndpoints xmi:id="NamedEndPoint_2" endPointName="SOAP_CONNECTOR_ADDRESS">
    <endPoint xmi:id="EndPoint_2" host="RCHASO2B" port="8880"/>
    </specialEndpoints>
  <specialEndpoints xmi:id="NamedEndPoint_3" endPointName="DRS_CLIENT_ADDRESS">
    <endPoint xmi:id="EndPoint_3" host="RCHASO2B" port="7873"/>
    </specialEndpoints>
      <specialEndpoints xmi:id="NamedEndPoint_6"
endPointName="SAS_SSL_SERVERAUTH_LISTENER_ADDRESS">
    <endPoint xmi:id="EndPoint_6" host="RCHASO2B" port="9501"/>

    <endPoint xmi:id="EndPoint_6" host="RCHASO2B" port="9501"/>
    </specialEndpoints>
      <specialEndpoints xmi:id="NamedEndPoint_7"
endPointName="CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS">
    <endPoint xmi:id="EndPoint_7" host="RCHASO2B" port="9503"/>
    </specialEndpoints>
      <specialEndpoints xmi:id="NamedEndPoint_8"
endPointName="CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS">
    <endPoint xmi:id="EndPoint_8" host="RCHASO2B" port="9502"/>
    </specialEndpoints>
   </serverEntries>
    <serverEntries xmi:id="ServerEntry_2" serverDisplayName="nodeagent"
serverName="nodeagent" serverType="NODE_AGENT">
  <specialEndpoints xmi:id="NamedEndPoint_15" endPointName="BOOTSTRAP_ADDRESS">
    <endPoint xmi:id="EndPoint_4" host="RCHASO2B" port="2809"/>
    </specialEndpoints>
 <specialEndpoints xmi:id="NamedEndPoint_4" endPointName="ORB_LISTENER_ADDRESS">
    <endPoint xmi:id="EndPoint_5" host="RCHASO2B" port="9900"/>
    </specialEndpoints>
      <specialEndpoints xmi:id="NamedEndPoint_5"
endPointName="SAS_SSL_SERVERAUTH_LISTENER_ADDRESS">
    <endPoint xmi:id="EndPoint_9" host="RCHASO2B" port="9901"/>
    </specialEndpoints>
</specialEndpoints>
      <specialEndpoints xmi:id="NamedEndPoint_9"
endPointName="CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS">
     <endPoint xmi:id="EndPoint_10" host="RCHASO2B" port="9101"/>
     </specialEndpoints>
      <specialEndpoints xmi:id="NamedEndPoint_10"
endPointName="CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS">
     <endPoint xmi:id="EndPoint_11" host="RCHASO2B" port="9201"/>
     </specialEndpoints>
```

Figure 7-18 serverindex.xml for server (node) default, example A - Part 1 of 2

```
 <specialEndpoints xmi:id="NamedEndPoint_11" endPointName="NODE_DISCOVERY_ADDRESS">
      <endPoint xmi:id="EndPoint_12" host="RCHASO2B" port="7272"/>
     </specialEndpoints>
      <specialEndpoints xmi:id="NamedEndPoint_12"
endPointName="NODE_MULTICAST_DISCOVERY_ADDRESS">
     <endPoint xmi:id="EndPoint_13" host="232.133.104.73" port="5000"/>
      </specialEndpoints>
<specialEndpoints xmi:id="NamedEndPoint_16" endPointName="DRS_CLIENT_ADDRESS">
      <endPoint xmi:id="EndPoint_14" host="RCHASO2B" port="7888"/>
    </specialEndpoints>
    <specialEndpoints xmi:id="NamedEndPoint_17" endPointName="SOAP_CONNECTOR_ADDRESS">
      <endPoint xmi:id="EndPoint_15" host="RCHASO2B" port="8878"/>
    </specialEndpoints>
    </specialEndpoints>
   </serverEntries>
    <serverEntries xmi:id="ServerEntry_3" serverDisplayName="JMSServer"
serverName="jmsserver" serverType="MESSAGE_BROKER">
  <specialEndpoints xmi:id="NamedEndPoint_18" endPointName="BOOTSTRAP_ADDRESS">
    <endPoint xmi:id="EndPoint_16" host="RCHASO2B" port="2810"/>
    </specialEndpoints>
      <specialEndpoints xmi:id="NamedEndPoint_13"
endPointName="JMSSERVER_DIRECT_ADDRESS">
    <endPoint xmi:id="EndPoint_17" host="RCHASO2B" port="5559"/>
    </specialEndpoints>
      <specialEndpoints xmi:id="NamedEndPoint_14"
endPointName="JMSSERVER_QUEUED_ADDRESS">
    <endPoint xmi:id="EndPoint_18" host="RCHASO2B" port="5558"/>
    </specialEndpoints>
 <specialEndpoints xmi:id="NamedEndPoint_19" endPointName="SOAP_CONNECTOR_ADDRESS">
    <endPoint xmi:id="EndPoint_19" host="RCHASO2B" port="8876"/>
    </specialEndpoints>
   </serverEntries>
 </serverindex:ServerIndex>
</xmi:XMI>
```

Figure 7-19 serverindex.xml for server (node) default, example A - Part 2 of 2

```
addNode RCHAS02B 8879 -instance webface -startingport 20300
CPF1010:  Subsystem name QEJBAS5 active.
ADMU0116I: Tool information is being logged in file
           /QIBM/UserData/WebAS5/Base/webface/logs/addNode.log
ADMU0001I: Begin federation of node RCHAS02B_webface with Deployment Manager at
           RCHAS02B:8879.
ADMU0009I: Successfully connected to Deployment Manager Server: RCHAS02B:8879
ADMU0505I: Servers found in configuration:
ADMU0506I: Server name: webface
ADMU2010I: Stopping all server processes for node RCHAS02B_webface
ADMU0510I: Server webface is now STOPPED
ADMU0024I: Deleting the old backup directory.
ADMU0015I: Backing up the original cell repository.
ADMU0012I: Creating Node Agent configuration for node: RCHAS02B_webface
ADMU0014I: Adding node RCHAS02B_webface configuration to cell: RCHAS02BNetwork
ADMU0016I: Synchronizing configuration between node and cell.
ADMU0018I: Launching Node Agent process for node: RCHAS02B_webface
ADMU0020I: Reading configuration for Node Agent process: nodeagent
ADMU0022I: Node Agent launched. Waiting for initialization status.
ADMU0030I: Node Agent initialization completed successfully. Process id is:
            023954/QEJBSVR/NODEAGENT
ADMU0300I: Congratulations: Your node RCHAS02B_webface has been successfully
            incorporated into the RCHAS02BNetwork cell.
ADMU9990I:
ADMU0306I: Be aware:
ADMU0302I: Any cell-level documents from the standalone RCHAS02B_webface
            configuration have not been migrated to the new cell.
ADMU0307I: You might want to:
ADMU0303I: Update the configuration on the RCHAS02BNetwork Deployment Manager
         with values from the old cell-level documents.
ADMU9990I:
ADMU0003I: Node RCHAS02B_webface has been successfully federated.

IBMWebSphereApplicationServer,Release5.0 WebSpherePluginConfigurationGenerator
Copyright IBM Corp., 1997-2002

PLGC0013I: Generating server plugin configuration file for all of the servers in cell
RCHAS02BNetwork.
PLGC0022W: No definition found for server dmgr.  Server will be ignored.

PLGC0020E: No valid server definitions found for server cluster
dmgr_RCHAS02BManager_Cluster. It will be ignored.
PLGC0022W: No definition found for server server1.  Server will be ignored.
PLGC0020E: No valid server definitions found for server cluster
server1_RCHAS02B_Cluster. It will be ignored.
PLGC0005I: Plugin Configuration file =
/QIBM/UserData/WebAS5/Base/webface/config/cells/plugin-cfg.xml
```

Figure 7-20 Messages to addNode server webface, example 1 - Part 1 of 2

```
ADCP0005I: Using cell RCHAS02BNetwork, node RCHAS02B_webface and server *ALL.
Display WAS instance:
  Instance name: webface
  Instance type: Federated Base Application Server
  Cell: RCHAS02BNetwork
  Node: RCHAS02B_webface

  Deployment manager for node: dmgr
    Deployment manager host: RCHAS02B
    Deployment manager SOAP port: 8879
    Deployment manager RMI port: 9809

  Information for server: webface
    Installed applications:
       DefaultApplication
       ivtApp
       WebFacing1

  Embedded JMS is not applicable to this server.

  Server status: Server is stopped.

    Ports in use:

         10400   Name service port
         10405   Soap port
         10404   Data replication service client port
         10406   SAS SSL server authentication listener port
         10408   CSIV2 server authentication listener port
         10407   CSIV2 mutual authentication listener port

 Information for server: nodeagent
   Installed applications:
     None

 Embedded JMS is not applicable to this server.

 Server status: Server is started. Job is 029075/QEJBSVR/NODEAGENT.
   Ports in use:
      20300   Name service port
      20308   Soap port
      20307   Data replication service client port
      20302   SAS SSL server authentication listener port
      20304   CSIV2 server authentication listener port
      20303   CSIV2 mutual authentication listener port
      20301   ORB listener port
      20305   Node discovery port
      20306   Node multicast discovery port
```

Figure 7-21 Messages to addNode server webface, example 1 - Part 2 of 2

7.3.3 Managing a node

After you federated a node to the cell, you need to learn the tools to manage that node. In this section we provide information about using the QShell scripts to manage a node. In 7.4, “Working with the ND administrative console” on page 298, we cover details of managing a cell by using the ND administrative console.

Changing a server in an ND environment via a script

For federated server in a cell, you have to make changes to the configuration for this server from the system which hosts the ND cell. This is because the master configuration repository is held by the cell, and the synchronization tasks override any changes made directly to a federated server.

You can use the `chgwassvr` script to make changes to your federated server from your iSeries server which hosts the Deployment Manager.

In this section we provide some examples of how to make changes to the following items:

- The Node Agent
- The JMS server
- The application server

For information about the syntax and the parameter for the `chgwassvr` script, see A.7.4, “Syntax and parameters for chgwassvr script” on page 534.

After any changes to the configuration as we do here, run the `syncNode` script or just wait for automatic synchronization. After the synchronization, the changes in the master configuration will be visible in the single node. Also, you can use the administrator console to synchronize a node manually; see 7.4.3, “Synchronizing configuration changes” on page 304. For details on how to use the `syncNode` script, see “Using the syncNode script” on page 290.

Note: In a Network Deployment environment, use the `chgwassvr` script from the Network Deployment instance to modify nodes in the cell. You have to use the -node parameter to specify the node name of the managed node containing the server that you want to modify.

Here are the steps to use the `chgwassvr` script:

1. On the OS/400 command line of the system where you run DM, enter the STRQSH (Start Qshell) command and press Enter.
2. On the Qshell command line, use the **cd** command to change to the Network Deployment directory:

   ```
   cd /QIBM/ProdData/WebAS5/ND/bin
   ```

3. Use the correct parameter values together with the `chgwassvr` script:

   Example 1: Change the ports of the Node Agent:

   We specify the ND instance name with the `-instance` parameter, the node name with the `-node` parameter and `nodeagent` for the `-server` parameter:

   ```
   chgwassvr -instance default -node RCHAS07_jmsmdb -server nodeagent -portblock 20100
   ```

   Figure 7-22 shows the result of our change.

```
chgwassvr -instance default -node RCHAS07_jmsmdb -server nodeagent -portblock 20100
ADCP0005I: Using cell RCHAS02BNetwork, node RCHAS07_jmsmdb and server nodeagent.
ADCP0001I: Ports changed successfully.
ADCP0002I: The following ports were changed to the specified value:
             Name service port:   20,100
             Soap port:   20,101
             Data replication service client port:   20,102
             SAS SSL server authentication listener port:   20,103
             CSIV2 server authentication listener port:   20,104
             CSIV2 mutual authentication listener port:   20,105
             ORB listener port:   20,106
             Node discovery port:   20,107
             Node multicast discovery port:   20,108
$
```

Figure 7-22 chgwassvr nodeagent tor jmsmdb server

Example 2: Change the ports to the application server jmsmdb. We specify the ND instance with -instance parameter, the node name for the -node parameter and application server name for the -server parameter:

```
chgwassvr -instance default -node RCHAS07_jmsmdb -server jmsmdb -nameservice 10420
```

Figure 7-23 shows the result of our change.

```
chgwassvr -instance default -node RCHAS07_jmsmdb -server jmsmdb -nameservice 10420
ADCP0005I: Using cell RCHAS02BNetwork, node RCHAS07_jmsmdb and server jmsmdb.
ADCP0001I: Ports changed successfully.
ADCP0002I: The following ports were changed to the specified value:
             Name service port:   10,420
```

Figure 7-23 chgwassvr application server jmsmdb tor jmsmdb server

4. After running the chgwasinst script, the configuration details for a corresponding managed process (application server, for example) have been changed. However, the changes exist only on the DM system. In order to copy them to the affected node, you have to perform a synchronization of the configuration data.

Using the syncNode script

The syncNode script allows you to manually synchronize the configuration between a single node and the deployment manager for the cell that the node belongs to. The master copy of the configuration documents for the node are copied from the deployment manager cell to the node.

Important: Before you run the syncNode script, the node agent and all application servers for the node must be stopped. You use the syncNode script on the server system where your node (instance) is running.

To run this script, your user profile must have *ALLOBJ authority.

To run the syncNode script, follow these steps:

1. If it is not already running, start the Deployment Manager instance.
2. Stop the node agent and all other managed processes (for example, application server and/or JMS server) for the node that you want to synchronize.

3. On the OS/400 command line of the system that you want to synchronize with DM, run the STRQSH (Start Qshell) command.
4. On the Qshell command line, use the **cd** command to change to the directory that contains the script:

   ```
   cd /QIBM/ProdData/WebAS5/Base/bin
   ```

5. Run the `syncNode` script:

   ```
   syncNode cell_host cell_port -instance instance
   ```

   Here, `cell_host` is the name of the server that hosts the Deployment Manager process, `cell_port` is the SOAP port number for the Deployment Manager, and `instance` is the name of the instance that represents the node that you want to synchronize.

   Examples:

   ```
   syncNode RCHAS02B 8879 -instance jmsmdb -restart
   ```

   This example synchronizes the configuration for the node where you run the *jmsmdb* instance with the Deployment Manager running on our RCHAS02B iSeries server. In this example, the Deployment Manager is running the default Network Deployment instance and has a SOAP port of 8879. The node agent is automatically started after the synchronization is completed, because we use the `-restart` parameter; see Figure 7-24 on page 291.

```
syncNode RCHAS02B 8879 -instance jmsmdb -restart
ADMU0116I: Tool information is being logged in file
           /QIBM/UserData/WebAS5/Base/jmsmdb/logs/syncNode.log
ADMU0401I: Begin syncNode operation for node RCHAS07_jmsmdb with Deployment
           Manager RCHAS02B: 8879
ADMU0016I: Synchronizing configuration between node and cell.
ADMU0018I: Launching Node Agent process for node: RCHAS07_jmsmdb
ADMU0020I: Reading configuration for Node Agent process: nodeagent
ADMU0022I: Node Agent launched. Waiting for initialization status.
ADMU0030I: Node Agent initialization completed successfully. Process id is:
           044296/QEJBSVR/NODEAGENT
ADMU0402I: The configuration for node RCHAS07_jmsmdb has been synchronized with
           Deployment Manager RCHAS02B: 8879
$
```

Figure 7-24 syncNode messages

Starting a node agent via startNode script

The `startNode` script starts the Node Agent process for a node that is part of a Network Deployment cell. Use this script to start a Node Agent from the system where your WAS instance is running.

The order of process startup needs to adhere to the following rules:

- A node agent can be running while the Deployment Manager is not, and vice versa. When the stopped process is started, discovery will occur automatically.
- The Deployment Manager can be running while a managed server is not, and vice versa.

Important: The application serving of a managed server is not dependent on the presence of a running Deployment Manager. The Deployment Manager is only required for making the permanent configuration changes (changes written to the master repository).

- Application servers can only be started if the local node agent is already started.

> **Note:** The node agent contains the Location Service Daemon (LSD) in which each application server registers itself on startup. If a node agent is restarted, then each application server on that node should also be restarted.
>
> The JMS server process of a Network Deployment node does not run an Object Request Broker (ORB) service, and does not need to register on startup with the node agent LSD. As such, the JMS server does not need to be restarted if the node agent is restarted.

Perform the following steps to start a Node Agent:

1. On the OS/400 command line, enter the STRQSH (Start Qshell) command.
2. On the Qshell command line, use the **cd** command to change to the directory that contains the script:

   ```
   cd /QIBM/ProdData/WebAS5/Base/bin
   ```

3. Invoke the `startNode` script:

   Example 1:

   ```
   startNode
   ```

   This command starts the node agent for the default instance.

   Example 2:

   ```
   startNode -instance webface
   ```

   This command starts the Node Agent for the `webface` instance (see in Figure 7-25).

```
startNode -instance webface
CPC1221: Job 030042/QEJBSVR/NODEAGENT submitted to job queue QEJBJOBQ in
library QEJBAS5.
EJB6123: Application server started.
 Cause. . . . . :   Application server nodeagent in Base instance webface
   has started and is ready to accept connections on admin port 9095.
$
```

Figure 7-25 startNode example

Verifying that the WebSphere Application Server node agent has started

The job name for the node agent process is NODEAGENT. If you are using multiple nodes of WebSphere Application Server and the servers have been added to a Network Deployment cell, you see one NODAGENT job for each instance.

When the node agent is ready for use, a message is written to the job log of the Node Agent job, indicating that the Node Agent is ready. The Node Agent runs on the iSeries system, where the WAS server that is managed by the WAS-ND runs. In this example, the WAS and the WAS-ND run on the same iSeries system.

To determine if the Node Agent is ready, perform these steps from an OS/400 command line:

1. Run the Work with Active Jobs (WRKACTJOB) command on an iSeries command line, specifying the QEJBAS5 subsystem on the subsystem (SBS) parameter:

   ```
   WRKACTJOB SBS(QEJBAS5)
   ```

   Press Enter.

2. We have two WAS instances running on our iSeries system, both of which are managed by the WAS-ND, and for both, the Node Agent is started; see Figure 7-26.

```
                          Work with Active Jobs                    RCHAS02B
                                                              12/05/02  02:20:54
CPU %:    1.0     Elapsed time:   00:00:53     Active jobs:   255

Type options, press Enter.
  2=Change   3=Hold   4=End   5=Work with   6=Release   7=Display message
  8=Work with spooled files   13=Disconnect ...

Opt  Subsystem/Job  User        Type  CPU %  Function        Status
     QEJBAS5        QSYS        SBS      .0                 DEQW
       ARTHUR       QEJBSVR     BCI      .4  PGM-QEJBSVR    JVAW
       JMSSAMPLE    QEJBSVR     BCI      .0  PGM-QEJBSVR    JVAW
       NODEAGENT    QEJBSVR     BCH      .0  PGM-QEJBSTRSVR JVAW
       NODEAGENT    QEJBSVR     BCH      .0  PGM-QEJBSTRSVR JVAW
                                                                  Bottom
Parameters or command
===>
F3=Exit   F5=Refresh       F7=Find      F10=Restart statistics
F11=Display elapsed data   F12=Cancel   F23=More options   F24=More keys
```

Figure 7-26 Active jobs in SBS QEJBAS5

3. If you are using multiple WebSphere Application Server nodes which have been added to a Network Deployment cell, you can determine which NODEAGENT job is for your specific server as follows:

   Specify option **5** (Work with Job) on the option line next to the nodeagent job, and press Enter.

   On the command line of the Work with Job display, specify option **10** (Display job log, if active), and press Enter.

   For this job you see here the iSeries program that is called to start the Node Agent and the parameter (PARM) that is used. The first message in the joblog contains the instance directory path and the name of the server. The last part of the value specified for the -instance parameter indicates the instance for which the nodeagent is running — in our case, that the server named arthur should be started.

```
                            Display Job Log
                                                              System:   RCHAS02B
Job . . :   NODEAGENT     User . . :   QEJBSVR       Number . . . :   033541

 >> CALL PGM(QEJBAS5/QEJBSTRSVR) PARM('-instance' '/QIBM/UserData/WebAS5/Base/
    arthur' '-server' 'nodeagent')









                                                                        Bottom
Press Enter to continue.

F3=Exit   F5=Refresh   F10=Display detailed messages   F12=Cancel
F16=Job menu           F24=More keys
```

Figure 7-27 PGM and parameter for start Node Agent

Press F10 (Display detailed messages) to display all messages.

Look for the message:

```
WebSphere application server application_server ready.
```

Here, *application_server* is in this case nodeagent.

If the message is not displayed, press F5 to refresh the job log messages until the message is displayed.

When you see a message similar to the one shown in Figure 7-28, the Node Agent is ready.

```
                          Display All Messages
                                                              System:   RCHAS02B
Job . . :   NODEAGENT     User . . :   QEJBSVR       Number . . . :   033541

 >> CALL PGM(QEJBAS5/QEJBSTRSVR) PARM('-instance' '/QIBM/UserData/WebAS5/Base/
    arthur' '-server' 'nodeagent')
    Server starting with user profile QEJBSVR and JDK 1.3.1.
    WebSphere application server nodeagent ready.


                                                                        Bottom
Press Enter to continue.

F3=Exit   F5=Refresh   F12=Cancel   F17=Top   F18=Bottom
```

Figure 7-28 Node Agent is ready

Set your cursor to the last line (WebSphere application server nodeagent ready) and press F1. You should see a message similar to one shown in Figure 7-29.

```
                        Additional Message Information

Message ID . . . . . :   EJB0106       Severity . . . . . :   00
Message type . . . . . :   Information
Date sent  . . . . . :   02/02/03      Time sent  . . . . . :   02:52:27

Message . . . . :   WebSphere application server nodeagent ready.
Cause . . . . . :   WebSphere application server nodeagent in job
  033541/QEJBSVR/NODEAGENT is ready to handle administrative requests on port
  9095.

  This is the admin port for the Deployment Manager process!

                                                                       Bottom
Press Enter to continue.

F3=Exit   F6=Print   F9=Display message details   F12=Cancel
F21=Select assistance level
```

Figure 7-29 Additional message information for node agent job for server arthur

Stopping a node agent via stopNode script

The stopNode script stops the Node Agent process for a node that is part of a Network Deployment cell. The stopNode script is available in the WebSphere Application Server product only. Use this script to stop a Node Agent from the node (system) where your WAS instance is running.

Perform the following steps to stop a Node Agent:

1. On the OS/400 command line, enter the STRQSH (Start Qshell) command.
2. On the Qshell command line, use the **cd** command to change to the directory that contains the script:

   ```
   cd /QIBM/ProdData/WebAS5/Base/bin
   ```

3. Invoke the stopNode script:

   Example:

   ```
   stopNode -instance arthur
   ```

   This example stops the node agent for the arthur instance.

```
stopNode -instance arthur
ADMU0116I: Tool information is being logged in file
           /QIBM/UserData/WebAS5/Base/arthur/logs/nodeagent/stopServer.log
ADMU3100I: Reading configuration for server: nodeagent
ADMU3201I: Server stop request issued. Waiting for stop status.
ADMU4000I: Server nodeagent stop completed.
$
```

Figure 7-30 stopNode arthur

Removing a node from the cell

You can use the administrative console or the `removeNode` script to remove application server nodes from a Network Deployment cell. When you remove a node, it becomes a stand-alone base application server. The applications that were part of the node remain in the Network Deployment cell, because such applications may subsequently be deployed on additional servers in the network deployment cell. You can remove a node from a Network Deployment instance by one of these methods:

- Use the administrative console
- Use the `removeNode` script

Remove a node via the administrative console

To use the administrative console to remove a node from a Network Deployment instance, follow these steps:

1. The Network Deployment instance from which you want to remove the node must be running.
2. Stop the WAS instance you want to remove from the Network Deployment cell.
3. Start the administrative console for your Network Deployment instance.
4. In the topology tree, expand **System Administration** and click **Nodes**.
5. On the Nodes page, select the node that you want to remove.
6. Click **Remove Node**.
7. Click **OK**.
8. **Save** the configuration.

The removeNode script

The `removeNode` script removes a node from a Network Deployment cell. After this process, the removed server is running as a standalone server in an WebSphere Application Server instance. The `removeNode` script only removes the node specific configuration from the cell.

Note: Depending on the size and location of the node you remove from the cell, this script can take a few minutes to complete.

To use the `removeNode` script to remove a node from a Network Deployment cell, follow these steps:

1. The Deployment Manager of your ND instance from which you want to remove the node must be running.
2. On the OS/400 command line, run the **STRQSH** (Start Qshell) command.
3. On the Qshell command line, use the **cd** command to change to the directory that contains the script:

   ```
   cd /QIBM/ProdData/WebAS5/Base/bin
   ```

4. Stop the server that you want to remove from the Network Deployment cell.
5. Run the `removeNode` script:

   ```
   removeNode -instance instanceName
   ```

   Here, `instanceName` is the name of the WAS instance that you want to remove from the cell. Typically, the name of the instance is the second part of the node name. For example, if the name of the node is `RCHASO2B_mark`, the name of the instance is `mark` (see Figure 7-31).

```
removeNode -instance mark
ADMU0116I: Tool information is being logged in file
           /QIBM/UserData/WebAS5/Base/mark/logs/removeNode.log
ADMU2001I: Begin removal of node: RCHAS02B_mark
ADMU0009I: Successfully connected to Deployment Manager Server:
           RCHAS02B.ITSO.IBM.COM:8879
ADMU0505I: Servers found in configuration:
ADMU0506I: Server name: mark
ADMU0506I: Server name: nodeagent
ADMU2010I: Stopping all server processes for node RCHAS02B_mark
ADMU0512I: Server mark cannot be reached. It appears to be stopped.
ADMU0510I: Server nodeagent is now STOPPED
ADMU2019I: Removing installed applications from node RCHAS02B_mark.
ADMU2021I: Removing all servers on this node from all clusters in the cell.
ADMU2018I: Node RCHAS02B_mark has been removed from the Deployment Manager
           configuration.
ADMU0526I: Deleting Queue Manager for node RCHAS02B_mark on server jmsserver
ADMU0528I: Details of Queue Manager deletion may be seen in the file:
           deleteMQ.RCHAS02B_mark_jmsserver.log
ADMU2014I: Restoring original configuration.
ADMU2017I: The local original configuration has been restored.
ADMU9990I:
ADMU0306I: Be aware:
ADMU2031I: Any applications that were uploaded to the RCHAS02BNetwork cell
           configuration during addNode using the -includeapps option are not
           uninstalled by removeNode.
ADMU0307I: You might want to:
ADMU2032I: Use wsadmin or the Administrative Console to uninstall any such
           applications from the Deployment Manager.
ADMU9990I:
ADMU2024I: Removal of node RCHAS02B_mark is complete.
$
```

Figure 7-31 Remove node mark

Cleanup node

The `cleanupNode` script removes a node configuration from the cell repository. Use this script to clean up a node only if you have a node defined in the cell configuration, but the node no longer exists. For example, if you use the `removeNode` script to remove a node, and you specify the `-force` parameter, the script removes the node even if it cannot connect to the deployment manager. If the script cannot connect to the deployment manager, the configuration information for the cell is not changed to indicate that the node was removed. The `cleanupNode` script removes the node information from the cell repository.

The `cleanupNode` script is available with WebSphere Application Server Network Deployment only. To run this script, your user profile must have *ALLOBJ authority.

To run the `cleanupNode` script, follow these steps:

1. On the OS/400 command line, run the STRQSH (Start Qshell) command.
2. On the Qshell command line, use the **cd** command to change to the directory that contains the script:

   ```
   cd /QIBM/ProdData/WebAS5/ND/bin
   ```

3. Run the `cleanupNode` script:

   ```
   cleanupNode node_name -instance instance
   ```

For example:

```
cleanupNode RCHAS02B_mark -instance mark
```

7.4 Working with the ND administrative console

The administrative console in a Network Deployment is a Web-based application that runs in a browser. It has a superset of functions available in the administrative console in a WAS environment.

The browser-based administrative console for WAS-ND requires that cookies are enabled in the browser, for more information see Chapter 6.7, “Working with the Administrative Console” on page 159.

To start the administrative console on a workstation, perform these steps:

1. Start the Deployment Manager (see “Starting Deployment Manager” on page 274).
2. Open this URL in your browser:

   ```
   http://your.server.name:port/admin
   ```

   Here, `your.server.name` is the host name of the iSeries server on which the Deployment Manager is running, and `port` is the port number as noted in the ready message in the joblog of the Deployment Manager; see “Verifying that the Deployment Manager has started” on page 275. For the default Network Deployment instance, the administrative console default port is 9090.

   In our environment we changed the administrative console port to 9095; see 7.3, “Building a WAS-ND cell” on page 270. So, in our case, the URL would be:

   ```
   http://rchas02b:9095/admin
   ```

3. When prompted, enter a user ID and click **OK**; see Figure 7-32 on page 299.

Note: The user ID does not need to be an OS/400 user profile. This user ID is used only to track which users make changes to the application server configuration. You can add security functions for the Administrative Console. For more information, see 8.2, “Enabling global security” on page 355.

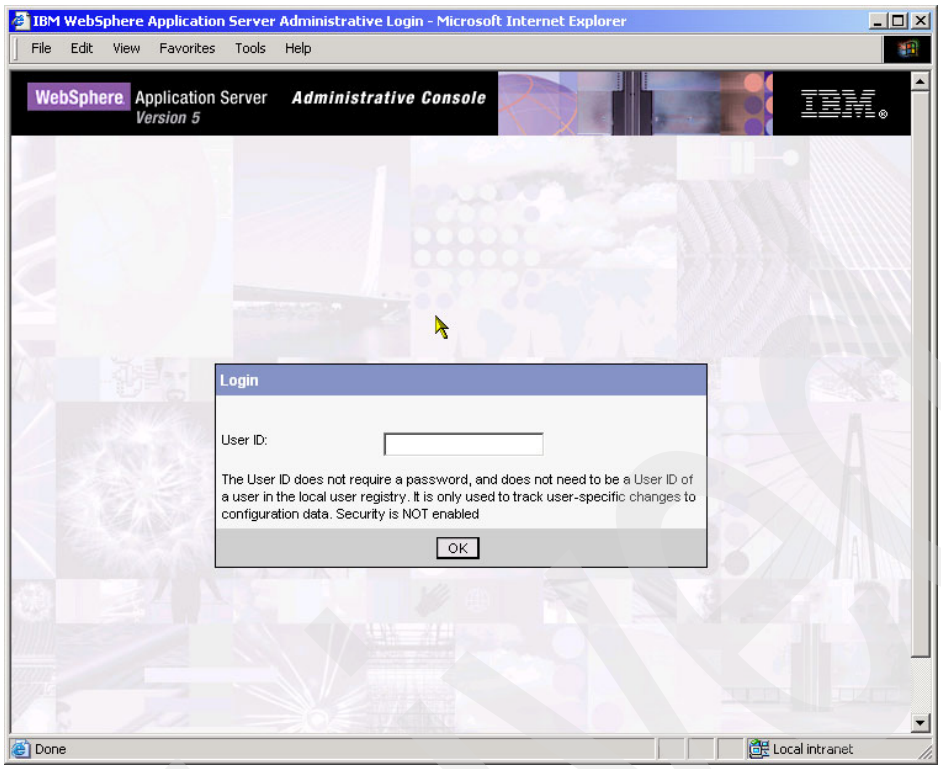


Figure 7-32 Start the administrative console from a browser

The WAS-ND administrative console comes up; see Figure 7-33. Notice the different components that are available in comparison to the administrative console in a WAS environment.

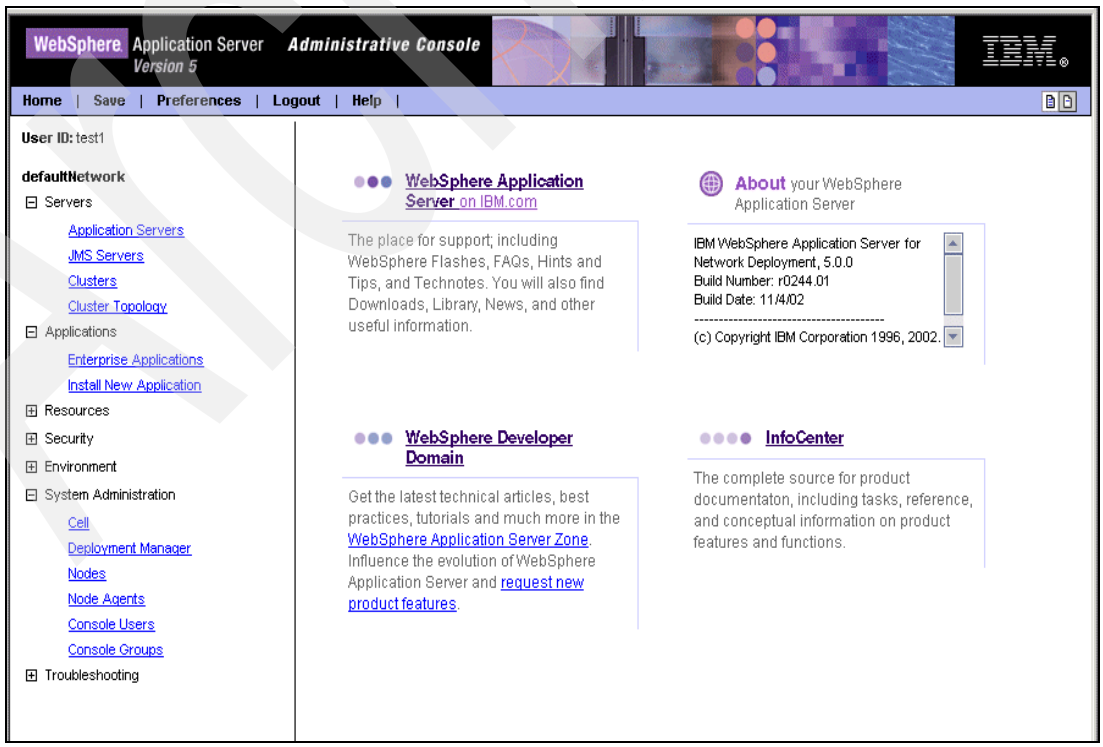


Figure 7-33 ND Administrative Console start

7.4.1 Updating a virtual host

In this section we describe the steps that are necessary to get our federated WAS server (node) running again as in the WAS environment. In our scenario we want to use the same URLs and HTTP ports in the new ND environment as we used in the standalone WAS environment. Also, we want to use the same several HTTP servers that we used before federating the nodes.

For more information about virtual host refer to Chapter 6.7.2, “Configuring a virtual host” on page 161.

Follow these steps to update the virtual host table:

1. Start the ND administrative console for your Deployment Manager instance; see 7.4, “Working with the ND administrative console” on page 298.
2. Expand **Environment** in the left frame of the administrative console.
3. Click **Virtual Hosts** in the left frame.
4. Click **default host** in the right frame.
5. Click **Host Aliases** in the Additional Properties section.

   You will see that there are only the default host aliases definitions after starting the ND environment and adding nodes to the ND cell via the `addNode` script, as shown in Figure 7-34.

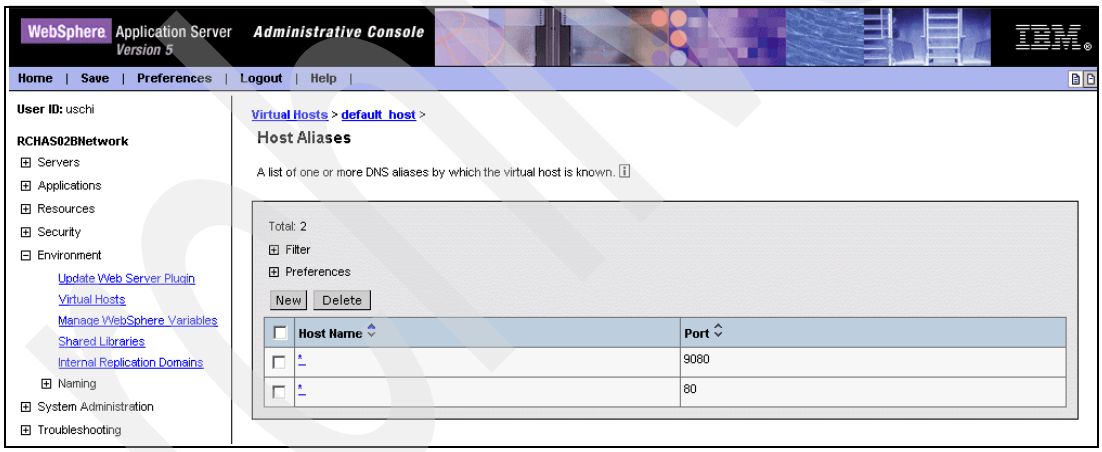


Figure 7-34 Host aliases in the ND environment

6. We need to add a new host alias for each federated server. Define the Host name and external HTTP port you want to use for your federated server. Do the following steps for every node you added to the ND cell:
   a. Click **New**.
   b. Define the Host name and port number

      In our example, for Host name we use ***** and for port number we use **10416** (see Figure 7-35).
   c. Click **OK**.

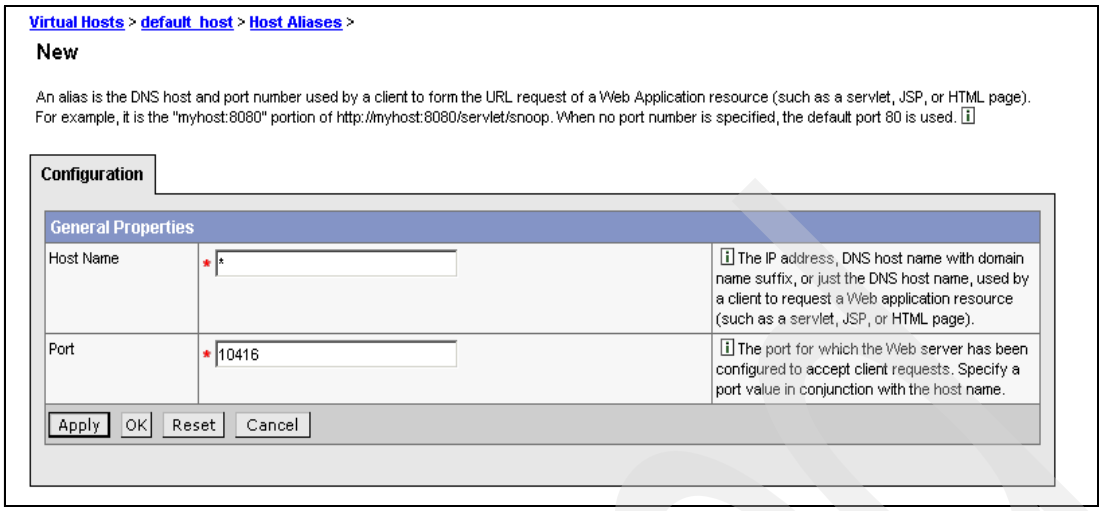


Figure 7-35 Add virtual host for example 1, the webface instance

Figure 7-36 shows our definitions in the default hosts Host Aliases display.

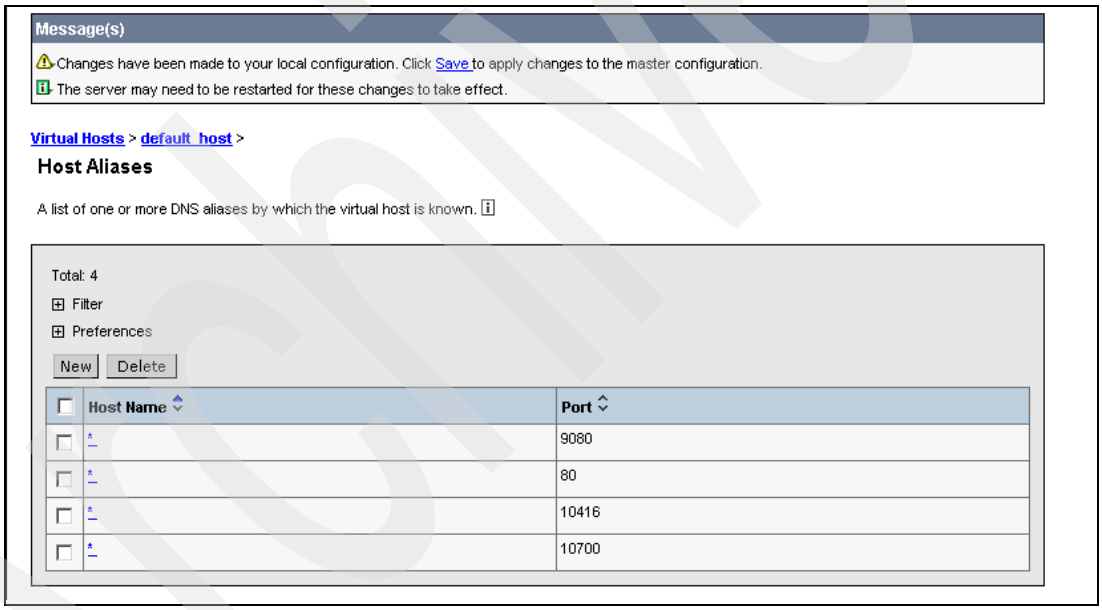


Figure 7-36 Virtual host definitions for our sample nodes, default, webface, and arthur

7. Click **Save** to save your configuration.
8. Click the **Save** button on a confirmation panel to save your workstation changes to the master configuration.

7.4.2 Updating the Web server plug-in configuration

The Web server plug-in configuration file (plugin-cfg.xml) specifies what URIs and host aliases can be served by the WAS environment.

The process of updating the plug-in file is the same as described in 6.7.4, “Updating Web server plug-in configuration” on page 171.

The difference in an ND environment is that the plugin-cfg.xml file is placed in the config directory of the ND Server instance. It is stored in the directory:

```
/QIBM/UserData/WebAS5/ND/instance_name/config/cells
```

This is located in the IFS of your iSeries server, where `instance_name` is the name of your DM’s instance. The plugin-cfg.xml file is copied to every server’s configuration repository during the next synchronization of the configuration data.

Attention: When regenerating the Web server plugin configuration file from the network deployment administrative console or network deployment `GenPluginCfg` script, an exception will occur if the plugin-cfg.xml file already exists in the network deployment's config/cells directory. Although the recommended method for generating the Web server plugin for network deployment configurations is to generate the file via the base install for each base instance, the error can be circumvented by deleting the existing plugin-cfg.xml file before regeneration.

The GenPluginCfg script

The `GenPluginCfg` script regenerates the plugin-cfg.xml file, which stores runtime configuration information for the Web server plugin. It is available in WebSphere Application Server and WebSphere Application Server Network Deployment. This script is very useful if for any circumstances the update Web server plug-in via the administrative console does not work in error situations.

The regenerated file is stored in one of the following directories:

- For WebSphere Application Server:

  ```
  /QIBM/UserData/WebAS5/Base/instance/config/cells
  ```

  Here, `instance` is the name of the WAS (Base) instance.

- For WebSphere Application Server Network Deployment:

  ```
  /QIBM/UserData/WebAS5/ND/instance/config/cells
  ```

  Here, `instance` is the name of your DM instance.

To run this script, your user profile must have *ALLOBJ authority. We show two examples of using the script:

- **Example1:** We generate the plugin-cfg file for all of the clusters (servers) in our cell on the RCHAS02B iSeries server.

  Perform the following steps:

  a. On the OS/400 command line, enter the STRQSH (Start Qshell) command and press Enter.

  b. On the Qshell command line, use the **cd** command to change to the ND directory that contains the script:

  ```
  cd /QIBM/ProdData/WebAS5/ND/bin
  ```

c. Invoke the GenPluginCfg script using following the syntax, where we are working with the default instance of the ND environment. If you work with an other instance name for your Network Deployment instance, specify also the -instance instancename parameter:

```
GenPluginCfg -cell.name NetworkDeploymentCell -instance instanceName
```

Note: You can omit the instance parameter for the default DM instance.

We type in for our scenario:

```
GenPluginCfg -instance default -cell.name RCHAS02BNetwork
```

Figure 7-37 shows the message in the QShell panel.

```
GenPluginCfg -instance default -cell.name RCHAS02BNetwork

IBM WebSphere Application Server, Release 5.0
WebSphere Plugin Configuration Generator
Copyright IBM Corp., 1997-2002

PLGC0012I: Generating server plugin configuration file using the cluster definitions in
cell RCHAS02BNetwork.

PLGC0020E: No valid server definitions found for server cluster ITSO Cluster1.
  It will be ignored.

PLGC0005I: Plugin Configuration file =
/QIBM/UserData/WebAS5/ND/default/config/cells/plugin-cfg.xml
```

Figure 7-37 Regenerate plugin-cfg for the cell

- **Example 2:** We generate the plugin-cfg file for the single server bankap on the RCHAS02B iSeries server.

  Perform the following steps:

  a. On the OS/400 command line, enter the STRQSH (Start Qshell) command and press Enter.

  b. On the Qshell command line, use the **cd** command to change to the Base directory that contains the script:

  ```
  cd /QIBM/ProdData/WebAS5/Base/bin
  ```

  c. Invoke the GenPluginCfg script using following the syntax:

  ```
  GenPluginCfg -cell.name BaseApplicationServerCell -node.name serverNode
  -server.name serverName
  ```

  We type in for our scenario:

  ```
  GenPluginCfg -instance bankap -cell.name RCHAS02BNetwork
  ```

  Figure 7-38 shows the message in the QShell panel.

```
GenPluginCfg -instance bankap -cell.name RCHAS02BNetwork
PLGC0005I: Plugin Configuration file =
/QIBM/UserData/WebAS5/Base/bankap/config/cells/plugin-cfg.xml
```

Figure 7-38 regenerate plugin-cfg for a server on the cell

7.4.3 Synchronizing configuration changes

When you make changes to the configuration of a managed node through the deployment manager's administrative console, you must synchronize the configuration with the managed node. If you do not synchronize the configurations, the nodes use their old configurations until the next synchronization occurs. You can configure the Node Agent for a managed node to request configuration synchronization automatically, or you can send a request from the Deployment Manager to the Node Agent to synchronize the configurations.

In the case of automatic synchronization, the synchronization interval specifies the number of minutes that elapse between synchronizations. The default is 1 minute. Increase the time interval to synchronize files less often. You can do this by changing the settings for the synchronization service via the Deployment Manager administrative console.

In this section we describe the following tasks:

- Changing settings for automatic synchronization
- Synchronizing the configuration for a node via the ND administrative console

Changing settings for automatic synchronization

To change the settings for automatic synchronization, follow these steps:

1. Start the ND administrative console for your Deployment Manager instance; see 7.4, “Working with the ND administrative console” on page 298.
2. Expand **System administration** and click **Node Agents**. You will get the Node agents panel, which lists all node agents available on your cell; see Figure 7-39.

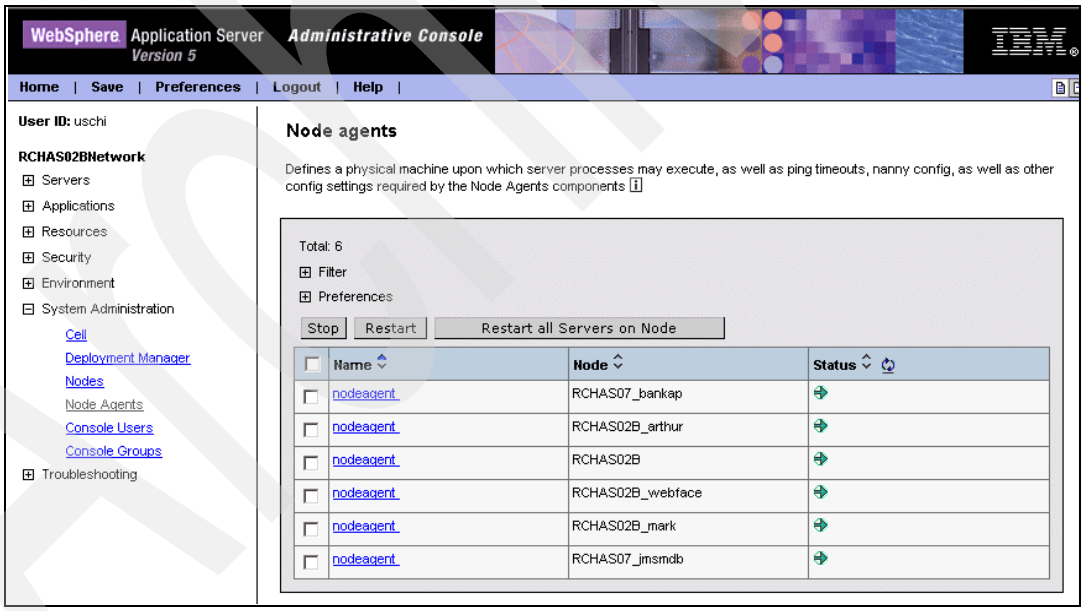


Figure 7-39 Node agents on our cell on RCHAS02B

3. Click the name of the Node Agent for which you want to change automatic synchronization.
4. In the next panel, click **File Synchronization Service**, you will get the next panel, as shown in Figure 7-40.
5. Increase the number of minutes that will elapse between synchronizations and click **OK**. The default value is 1 minute. In a production environment without frequent configuration changes, you should use a higher value.

> **Note:** This value is not used if automatic synchronization is disabled (see the **Automatic synchronization** checkbox).

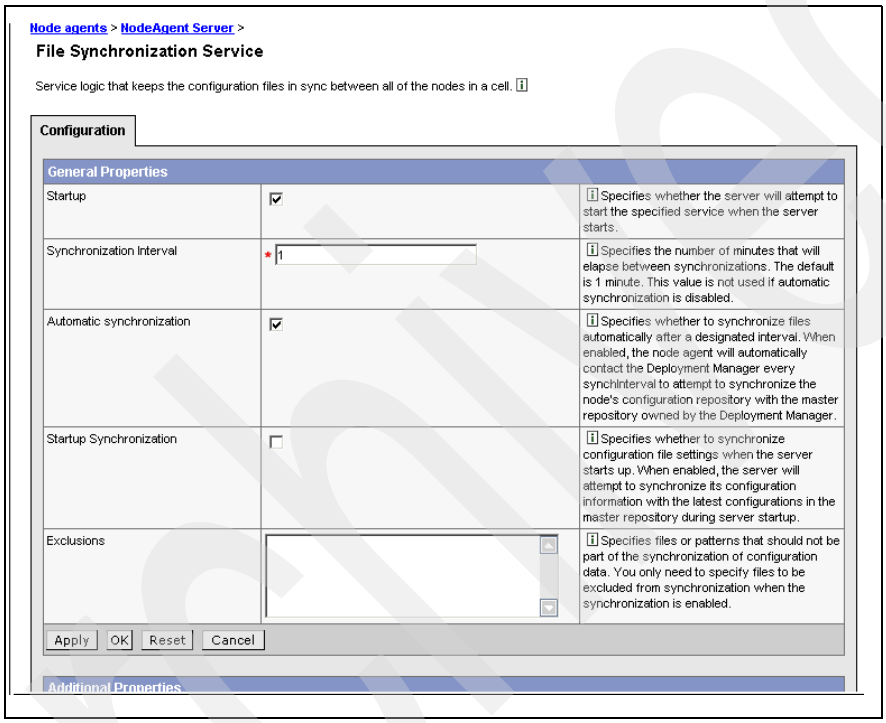


Figure 7-40 File synchronization service panel

6. **Save** the configuration.

Synchronizing the configuration via the ND administrative console

To synchronize the configuration for a node, follow these steps:

1. Start the ND administrative console for your Deployment Manager instance; see 7.4, “Working with the ND administrative console” on page 298.
2. Expand **System administration** and click **Nodes**. You will get the Nodes panel, as shown in Figure 7-41. It lists all the nodes in the cell and the status of the synchronization.

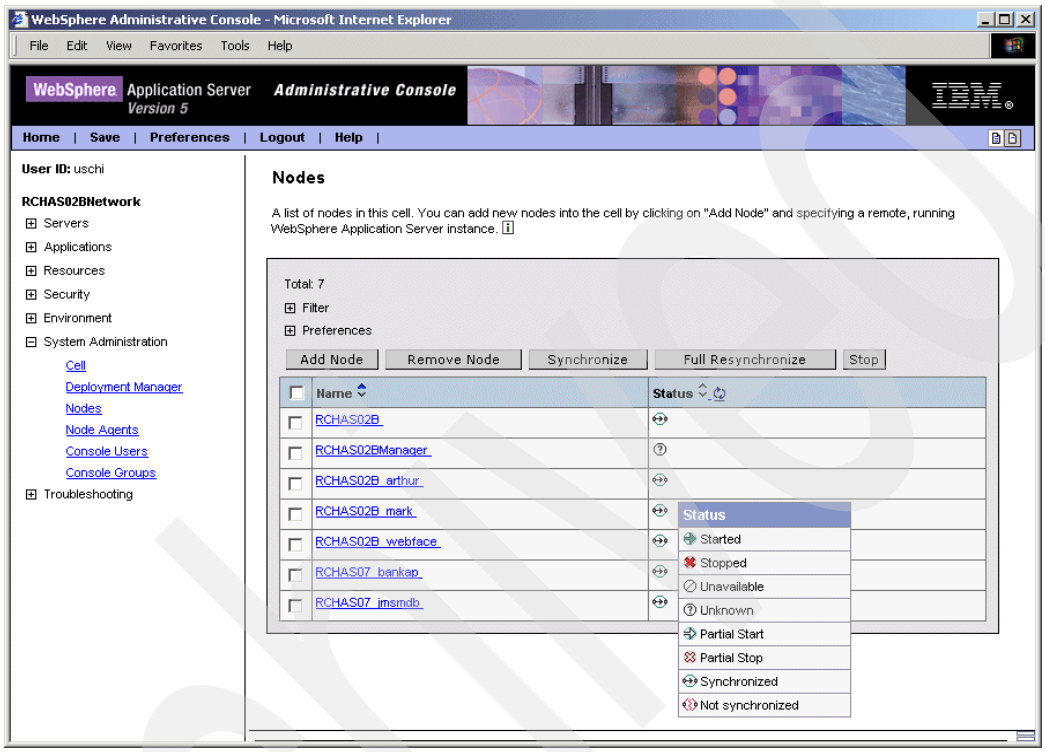


Figure 7-41 Nodes panel that lists our nodes on the cell on RCHAS02B

3. Click the checkbox for the node for which you want to synchronize the configuration data.
4. Click **Full Resynchronize**.

7.4.4 Displaying the managed processes of a node

To see the managed processes for a node, do the following steps:

1. Start your ND administrative console.
2. Expand **System Administration**.
3. Click **Nodes**.
4. From the Nodes panel (see Figure 7-41), click the node for which you want to see all managed processes. You will get the next panel, similar to Figure 7-42.

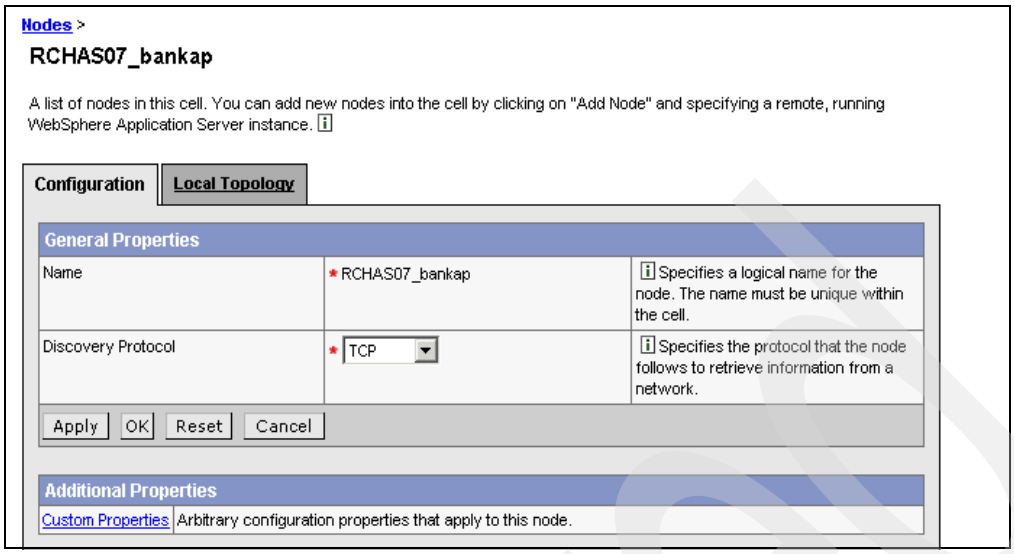


Figure 7-42 Properties for our sample node bankap on RCHAS07

5. Click the **Local Topology** tab.
6. Expand **servers** and you will see all managed processes (application servers) for the node, as shown in Figure 7-43.

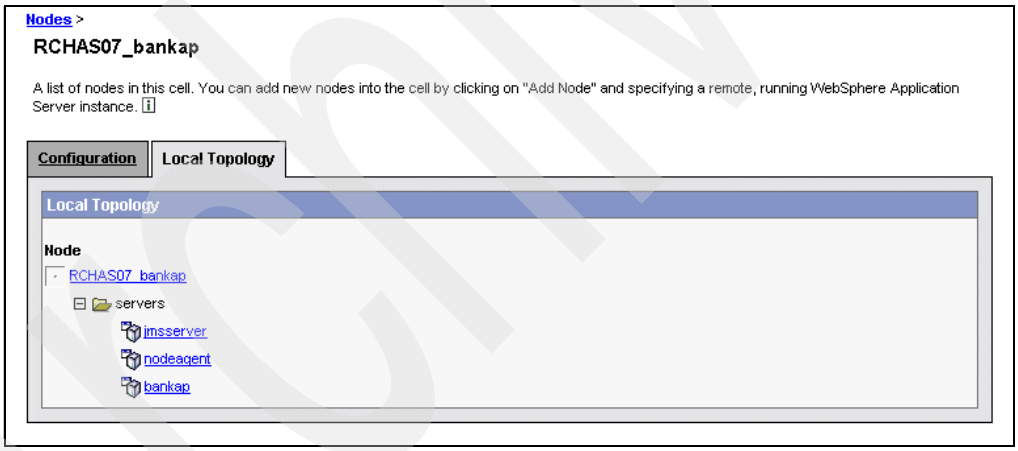


Figure 7-43 Servers for node RCHAS07_bankap

7.4.5 Adding a node via the administrative console

We recommend that you always use the `addNode` script, because the administrative console doesn't provide a way of specifying the user-defined ports, so the default ports are assigned to the nodeagent and JMS server. For details on how to use the `addNode` script, see 7.3.2, "Adding a node to the network deployment instance" on page 280.

To add a node to the cell via the administrative console, follow these steps:

Note: The application server for the instance that you want to add must be running.

In the following example we add the jmsmdb instance on RCHAS07 to the cell via the administrative console.

Important: We do this only to show what will happen. Please use this approach *only* when you want to add your first node to a cell.

1. Start the administrative console in the ND environment; see 7.4, “Working with the ND administrative console” on page 298.
2. In the topology tree, expand **System Administration** and click **Nodes**.
3. On the Nodes page, click **Add Node**.
4. On the Add Node page, specify the following information (see Figure 7-44):
   - **JMX Connector Type**: This field specifies the type of connector that the deployment manager uses to connect to the instance where the server resides that you want to add. We use the **SOAP** connector type.
   - **Host**: This field specifies the TCP/IP name of iSeries that hosts the server you want to federate. In our case it is **RCHAS07**.
   - **JMX Connector Port:** This field specifies the instance's SOAP or RMI port, that you want to add. This port number corresponds to the connector type that you specify in the JMX Connector Type field. Use the `dspwasinst` script from QShell of your federated node; see Figure 7-45, to find out the port numbers for the server in an instance you want to add. In our example the SOAP port for the instance jmsmdb is **10410**.
   - **Include Applications**: Select this option if you want to include the target server's applications in the new cell.

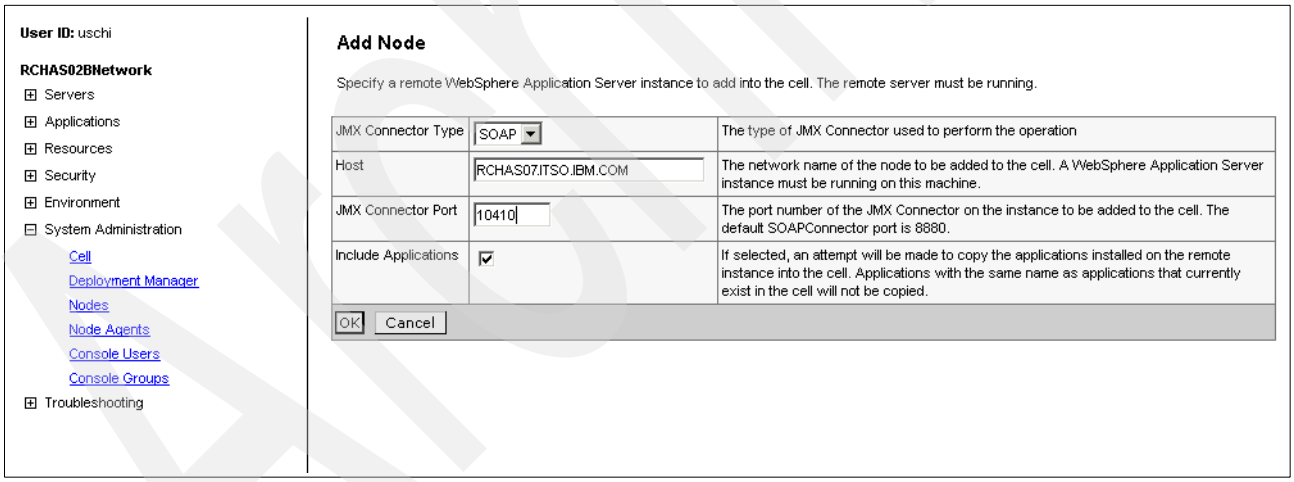


Figure 7-44 Add Node via administrator console

5. Click **OK.**

Note: This process may take several minutes to complete.

```
dspwasinst -instance jmsmdb
ADCP0005I: Using cell RCHAS07_jmsmdb, node RCHAS07_jmsmdb and server *ALL.
Display WAS instance:
  Instance name: jmsmdb
  Instance type: Base Application Server
  Cell: RCHAS07_jmsmdb
  Node: RCHAS07_jmsmdb

  Information for server: jmsmdb
    Installed applications:
      DefaultApplication
      ivtApp
      adminconsole
      MDBSamples

 Ports in use:
      10402   Administrative console port
      10414   Administrative console SSL-enabled port
      10405   Name service port
      10410   Soap port
      10401   Internal HTTP port
      10409   Data replication service client port
      10406   Internal Java Message Service server port
      10407   Queued Java Message Service server port
      10408   Direct Java Message Service server port
      10411   SAS SSL server authentication listener port
      10413   CSIV2 server authentication listener port
      10412   CSIV2 mutual authentication listener port
```

Figure 7-45 dspwasinst instance jmsmdb before adding the node

6. See the messages shown in the administrative console, Figure 7-46.

   Because we cannot define the `startingport` parameter, the default ports for the Node Agent and the JMS server are assigned. In our case this node is not the first one we add to the cell, so we have port conflicts and the Node Agent fails to start.

   In the /QIBM/UserData/WebAS5/Base/jmsmdb/logs/nodeagent/Systemout.log we find the following messages:

   ```
   Unable to start bootstrap server using port 2809. Verify that no servers or other
   processes are already using the bootstrap server port.Server nodeagent failed to start.
   Error occurred during startup.
   ```

7. In this situation we have to change:

   – The ports for the Node Agent
   – The ports for the JMS server

   To change these parameters; see “Changing a server in an ND environment via a script” on page 289.

```
ADMU0001I: Begin federation of node RCHAS07_jmsmdb with Deployment Manager at
RCHAS02B.ITSO.IBM.COM:8879.
ADMU0001I: Begin federation of node RCHAS07_jmsmdb with Deployment Manager at
RCHAS02B.ITSO.IBM.COM:8879.
ADMU0009I: Successfully connected to Deployment Manager Server:
RCHAS02B.ITSO.IBM.COM:8879
ADMU0505I: Servers found in configuration:
ADMU0506I: Server name: jmsmdb
ADMU2010I: Stopping all server processes for node RCHAS07_jmsmdb
ADMU0510I: Server jmsmdb is now STOPPED
ADMU0024I: Deleting the old backup directory.
ADMU0015I: Backing up the original cell repository.
ADMU0012I: Creating Node Agent configuration for node: RCHAS07_jmsmdb
ADMU0120I: DefaultApplication.ear will not be uploaded since it already exists in the
target repository.
ADMU0120I: ivtApp.ear will not be uploaded since it already exists in the target
repository.
ADMU0014I: Adding node RCHAS07_jmsmdb configuration to cell: RCHAS02BNetwork
ADMU0016I: Synchronizing configuration between node and cell.
ADMU0018I: Launching Node Agent process for node: RCHAS07_jmsmdb
ADMU0020I: Reading configuration for Node Agent process: nodeagent
ADMU0022I: Node Agent launched. Waiting for initialization status.
ADMU0031E: Node Agent launched but failed initialization. Process id was:
043870/QEJBSVR/NODEAGENT
ADMU0523I: Creating Queue Manager for node RCHAS07_jmsmdb on server jmsserver
ADMU0525I: Details of Queue Manager creation may be seen in the file:
createMQ.RCHAS07_jmsmdb_jmsserver.log
ADMU9990I:
ADMU0300I: Congratulations! Your node RCHAS07_jmsmdb has been successfully incorporated
into the RCHAS02BNetwork cell.
ADMU9990I:
ADMU0306I: Be aware:
ADMU0302I: Any cell-level documents from the standalone RCHAS07_jmsmdb configuration
have not been migrated to the new cell.
ADMU0307I: You might want to:
ADMU0303I: Update the configuration on the RCHAS02BNetwork Deployment Manager with
values from the old cell-level documents.
ADMU9990I:
ADMU0003I: Node RCHAS07_jmsmdb has been successfully federated.
The new node will not be available in the console until you log in again
Logout from the WebSphere Administrative Console
```

Figure 7-46 Messages in Administrative console - example add node jmdmdb

7.4.6 Managing node agents via the ND admin console

The ND administration console provides the following tasks to manage node agents:

- Stop a node agent
- Restart a node agent
- Restart all servers on node
- Change additional properties of a node agent

Note: You cannot start a node agent via the ND administrative console; therefore, you need to use the `startNode` script. See “Starting a node agent via startNode script” on page 291.

To manage a node through the administrative console of WAS-ND, follow these instructions:

1. Start the administrative console in the ND environment; see “Working with the ND administrative console” on page 298.
2. In the topology tree, expand **System Administration** and click **Node Agents**.

   You will get the Node Agents panel, which lists the node agents for all your federated WAS servers, as shown in Figure 7-47.

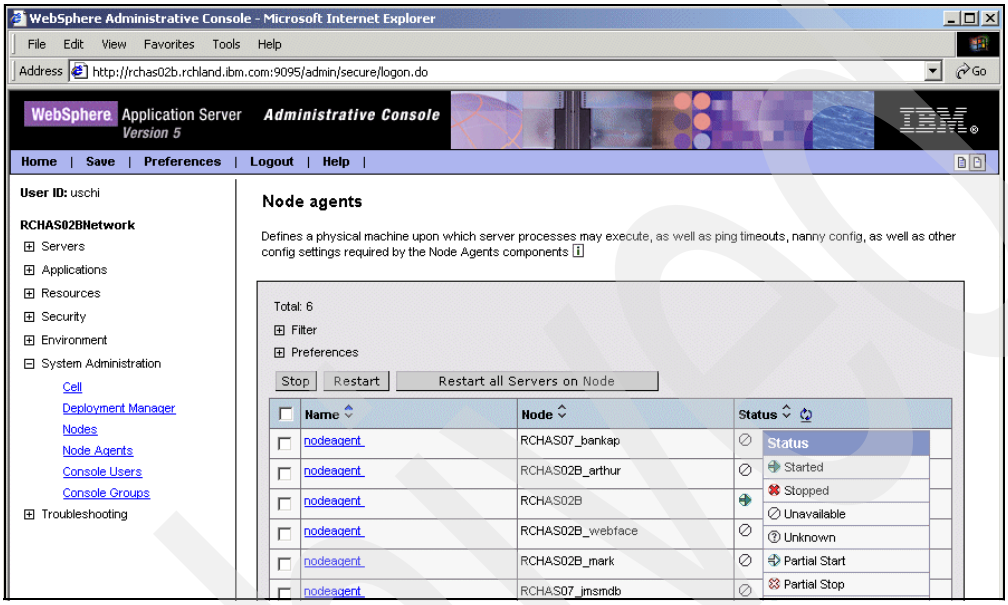


Figure 7-47 Work with Node agents via the ND administrative console

3. Select the check box for the node agent that you want to manage.
4. Click the appropriate button to:
   - Stop node agents
   - Restart node agents
   - Restart all servers on node

   Note: Restarting all servers on a node stops all processes (all servers) in the node and restarts the only servers that have their “Node restart state” parameter set to *star*”. The default setting for this parameter is *stopped*. For details on how to change this setting, see “Changing the node restart state for a server in a node” on page 313.

   If you do not change this parameter for the servers in a node, you have to start the processes (servers) as described in 7.4.8, “Starting/stopping an application server via the admin console” on page 314.

5. Change additional properties by clicking the link for the node agent you want to change. You will get the NodeAgent Server panel, as shown in Figure 7-48, for the chosen node agent. Here you can modify components for the services shown.

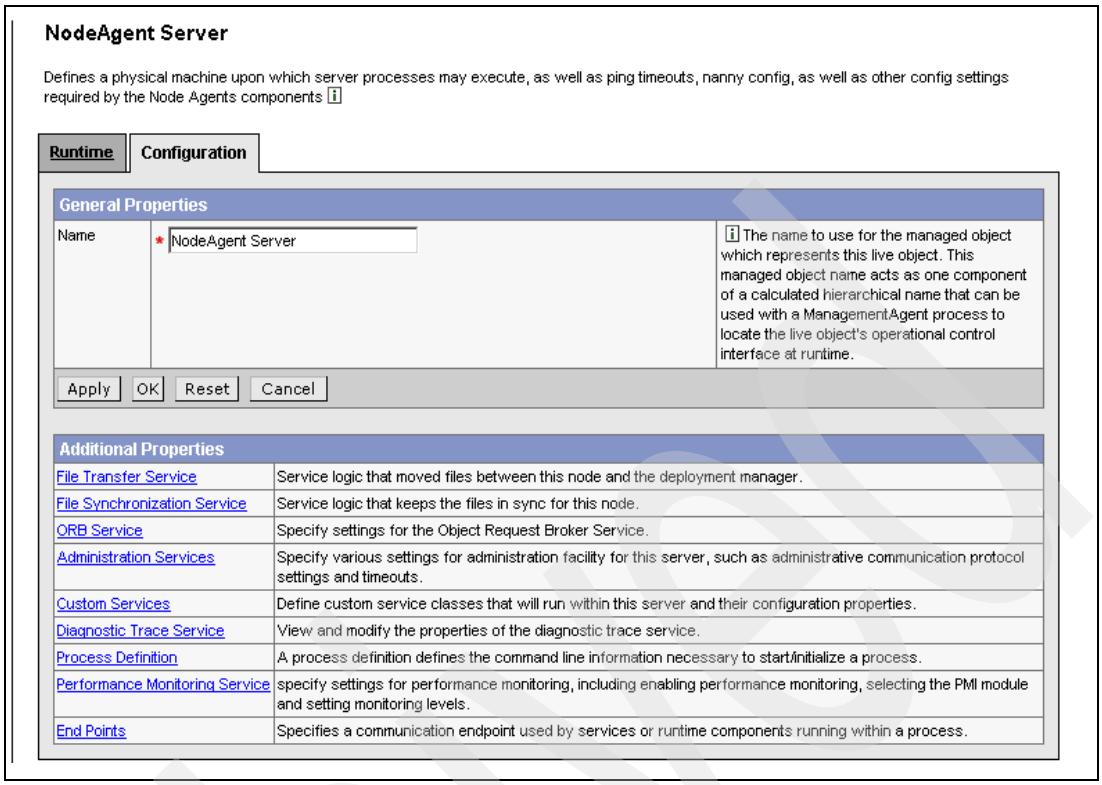


Figure 7-48 Additional Properties for node agents

6. As an example, here we show the properties for the Administration Services. We click the **Administration Services** link and get the Administration Servers properties panel, as shown in Figure 7-49. Here we can update the preferred JMX Connector type.

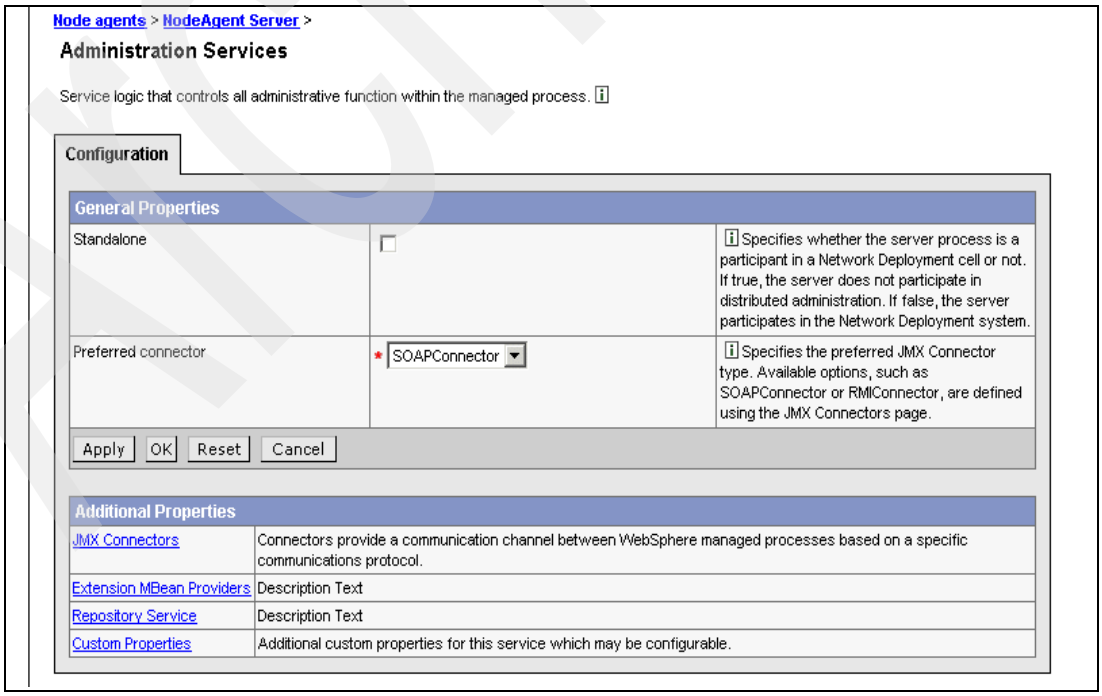


Figure 7-49 Administration Services properties for node agent webface

7.4.7 Changing the node restart state for a server in a node

Each application server has a parameter that defines the way this server behaves during an automatic restart. The *node restart state* parameter of a server is set to `stopped` by default.

To change this parameter of a server in that way, so that this server will be restarted when you restart a Node Agent via the administrative console, perform the following steps:

1. Start the administrative console in the ND environment; see 7.4, “Working with the ND administrative console” on page 298.
2. In the topology tree, expand **Server** and click **Application Servers**.
3. Click the link for the server you want to change.
4. Click **Process Definition**.
5. Click **Monitoring Policy**. You will get the Monitoring Policy panel, as shown in Figure 7-50.

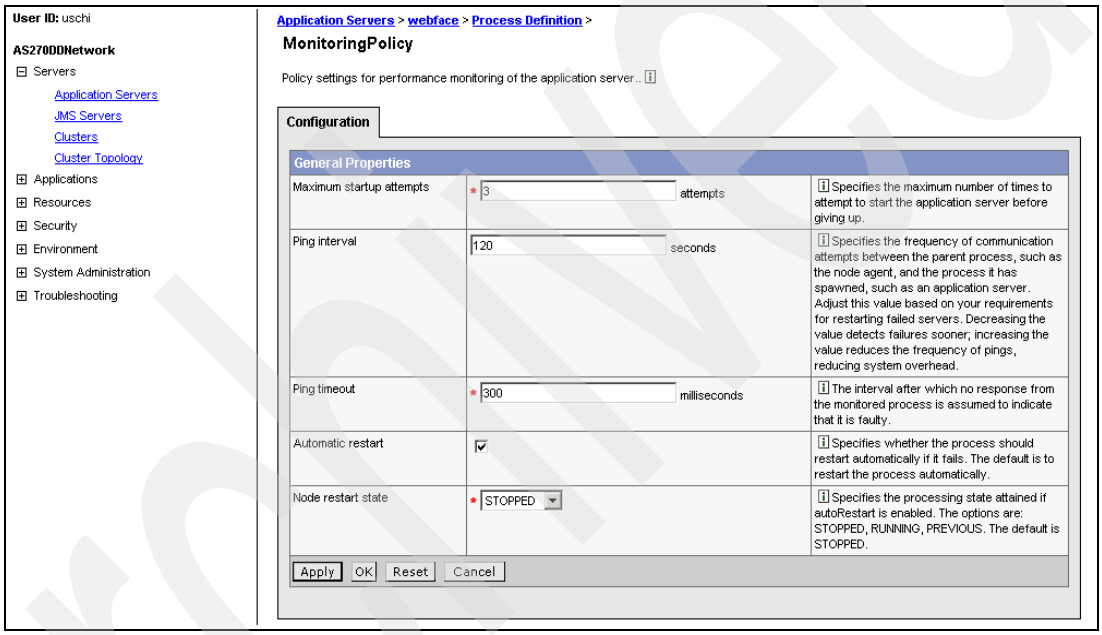


Figure 7-50 Default Node restart state - stopped

6. Set the Node restart state to **RUNNING**; see Figure 7-51.

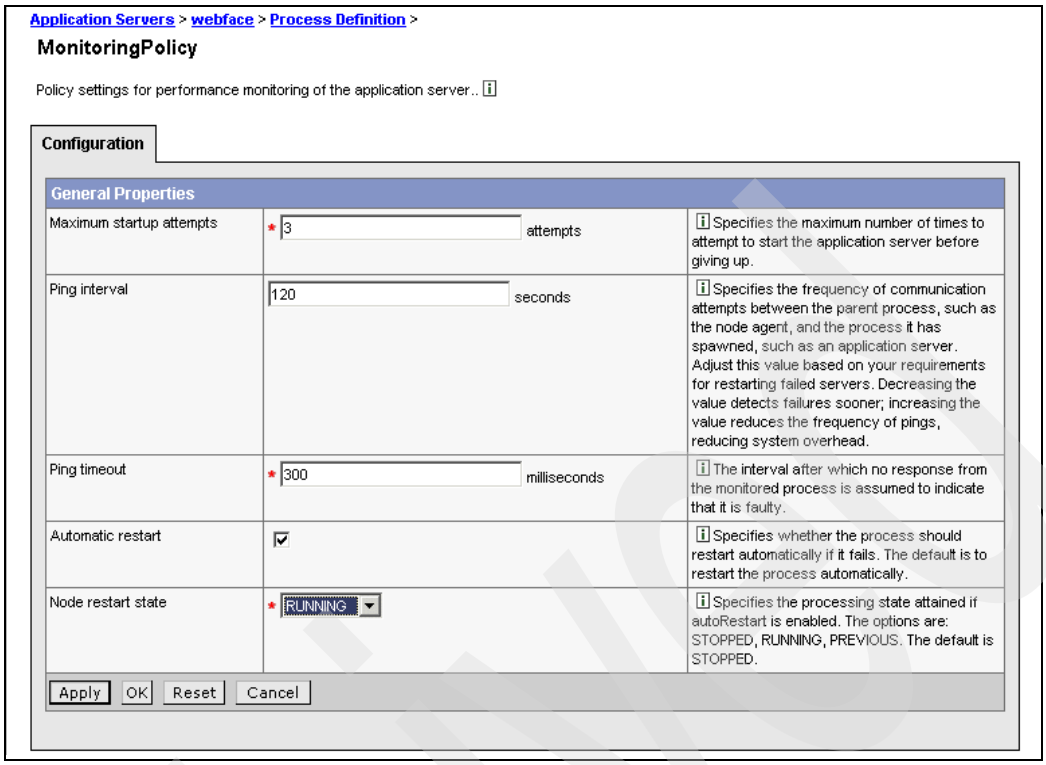


Figure 7-51 Node restart state- running

7. Click **Apply**. Notice that we also changed the **Ping interval** from 60 (default) to 120 seconds; see Figure 7-51.
8. **Save** the configuration change.

7.4.8 Starting/stopping an application server via the admin console

You can start or stop an application server of a federated node in two ways:

- Using the `startServer` or `stopServer` script from the iSeries server where your instance is running
- Using the ND administrative console

Note: Before you can start a federated server, the node agent must be running. This is true for both of the above-mentioned possibilities to start a server. So, start the node agent first.

To use the administrative console for starting or stopping an application server, follow these steps:

1. Start the administrative console in the ND environment; see 7.4, “Working with the ND administrative console” on page 298.
2. In the topology tree, expand **Servers** and click **Application Servers**. You will get the Application Servers panel, which lists the application servers for all your federated servers, as shown in Figure 7-52.

Figure 7-52 Application servers on cell RCHAS02B

3. Select the check box for the server that you want to start or stop and click the appropriate button (**Start** or **Stop**).

7.5 Changing a server via the administrative console

You can manage the properties of an application server with the WAS-ND administrative console. In this section we show how you can change a TCP/IP port for a server (the different services are called End Points in the administrative console):

1. Start the administrative console in the ND environment; see 7.4, “Working with the ND administrative console” on page 298.
2. In the topology tree, expand **Servers** and click **Application Servers**.
3. Click the link for the server you want to change.
4. Click **End Points** (you may need to scroll). You will get a panel similar to Figure 7-53.

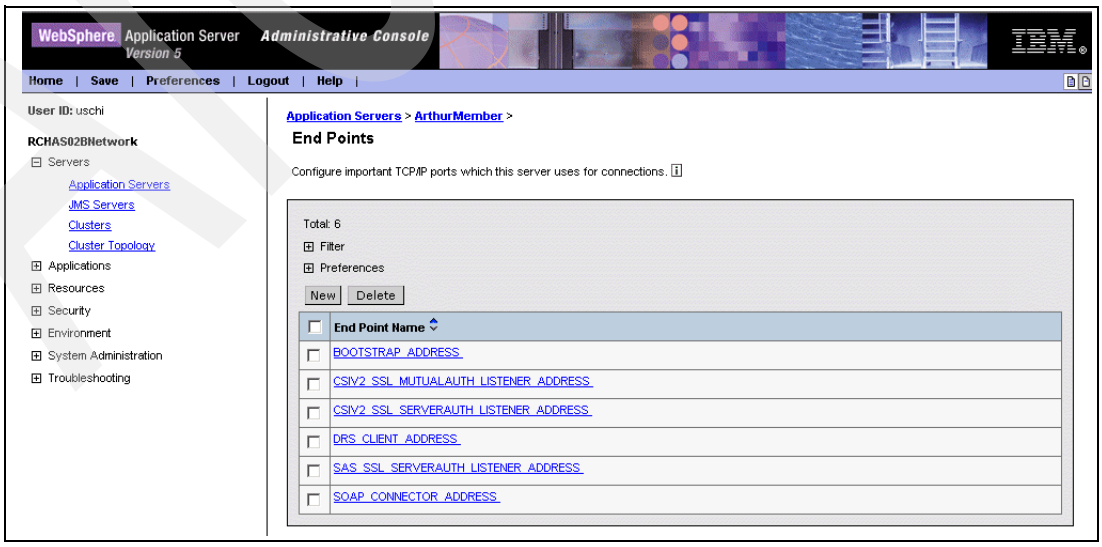


Figure 7-53 server End Points

5. Click the link for the service (**End Point Name)** you want to change.

> **Note:** The TCP/IP port on which the name service listens is called BOOTSTRAP ADDRESS in the administrative console environment.
>
> When you use the QShell scripts like `dspwasinst` or `chgwassvr`, the same TCP/IP port is called “Name service port”.

We want to change the TCP/IP port defined for the “name service port”, so we click **BOOTSTRAP ADDRESS** (see Figure 7-54).

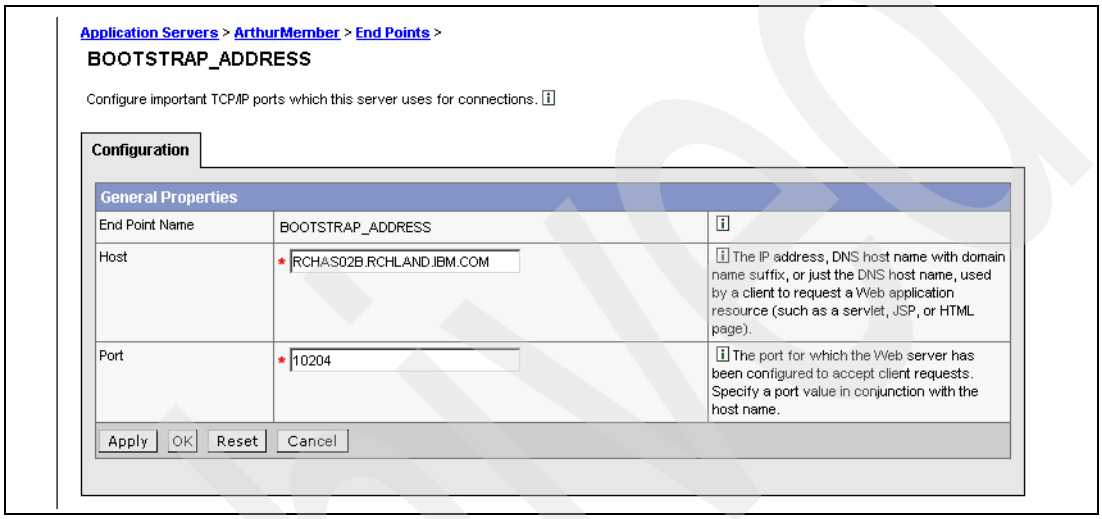


Figure 7-54 Name service port

6. Change the **Port** parameter to a different value.
7. Click **OK**.
8. **Save** your configuration.

7.5.1 Finding and configuring JMS resources in the ND environment

After federating a node to a cell, the resources defined in the federated server are also available in the ND environment. Depending on the scope you set while creating the resources in the application server, you can find them at the different scope levels.

In general, you can manage your JMS resources in the ND environment in the same way as in a WAS environment (refer to “JMS administered objects” on page 222 for further information). Only starting and stopping of an JMS server is different; see “Starting and stopping a JMS server in the ND environment” on page 319.

Following this sample, you can find your already defined JMS resources for a federated server:

1. Start the administrative console for the ND environment.
2. Expand **Resources** and click **WebSphere JMS Provider**.
3. In the WebSphere JMS Provider panel, the scope will be set to **Cell level** (see Figure 7-55).

   Follow the instructions described in the right frame beside the level input fields. Because we defined the resources for our example with the `jmsmdb` server at the node level (see “Configuring your JMS connection factory” on page 228), these definitions, after federating this server to the cell, are also at node level.

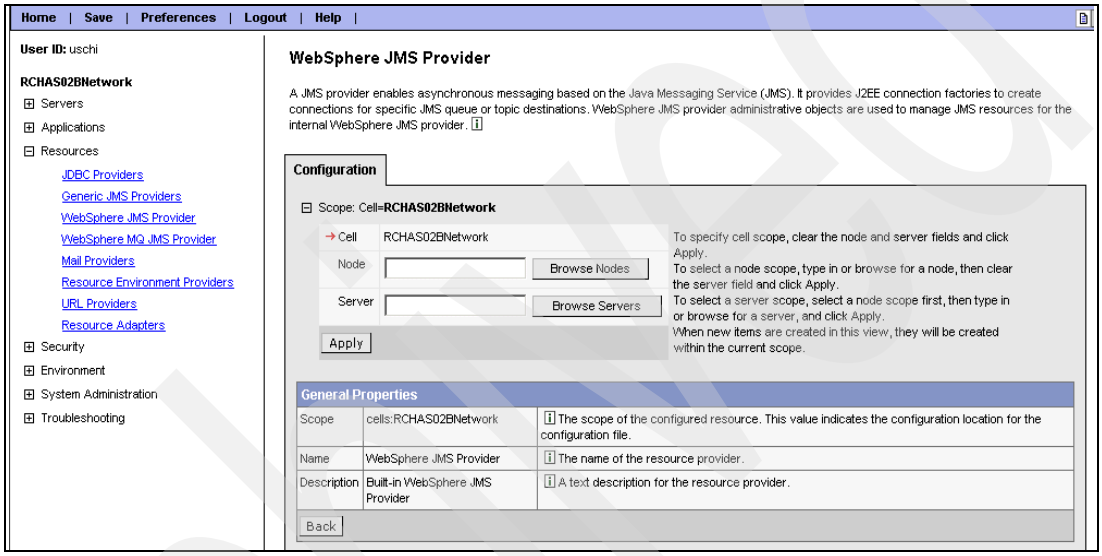


Figure 7-55 Set the scope to the node

4. The best way to change to the node level is to click the **Browse Nodes** button and select the node at which you want to search for available JMS resources (see Figure 7-56).
5. Click **OK**.

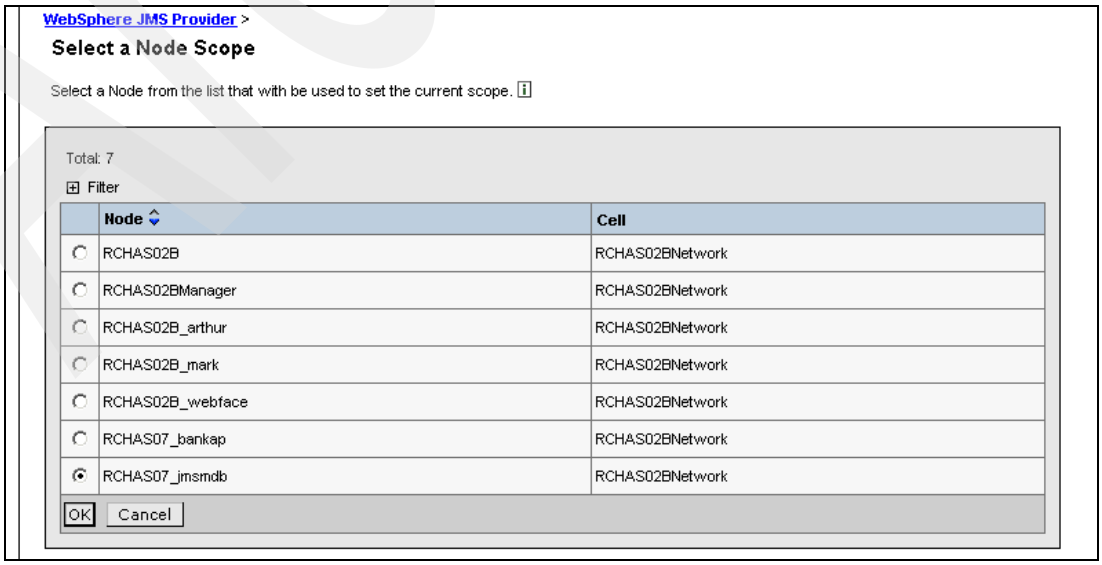


Figure 7-56 Select a Node Scope

6. Back in the WebSphere JMS Provider panel, click the **Browse Servers** button and select the server at which you want to search for the resources.
7. Click **OK** (see Figure 7-57).

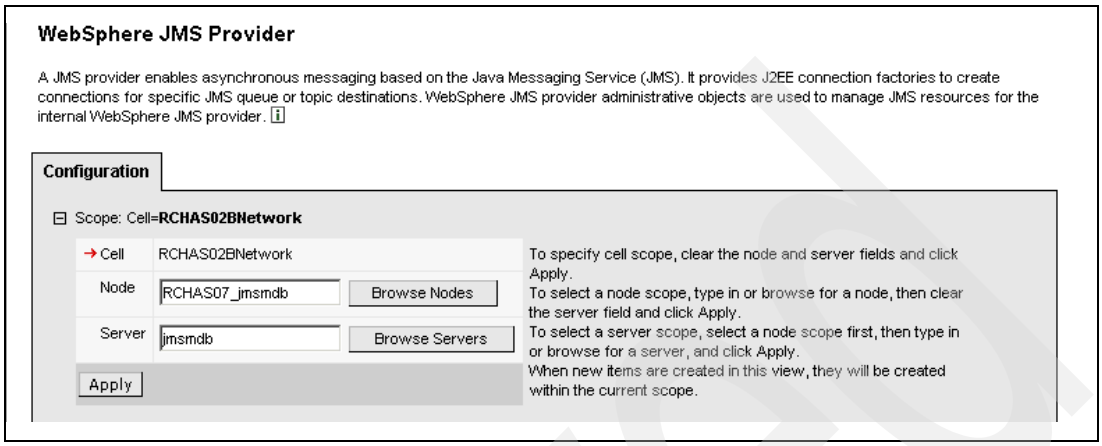


Figure 7-57 selected server for scope

8. Back in the WebSphere JMS Provider panel, clear the Server input field and click the **Apply** button. That will set the scope to the node level.
9. You can now select the resources that are already defined. In our case, for example, we select the **WebSphere Topic Connection Factories**; see Figure 7-58.

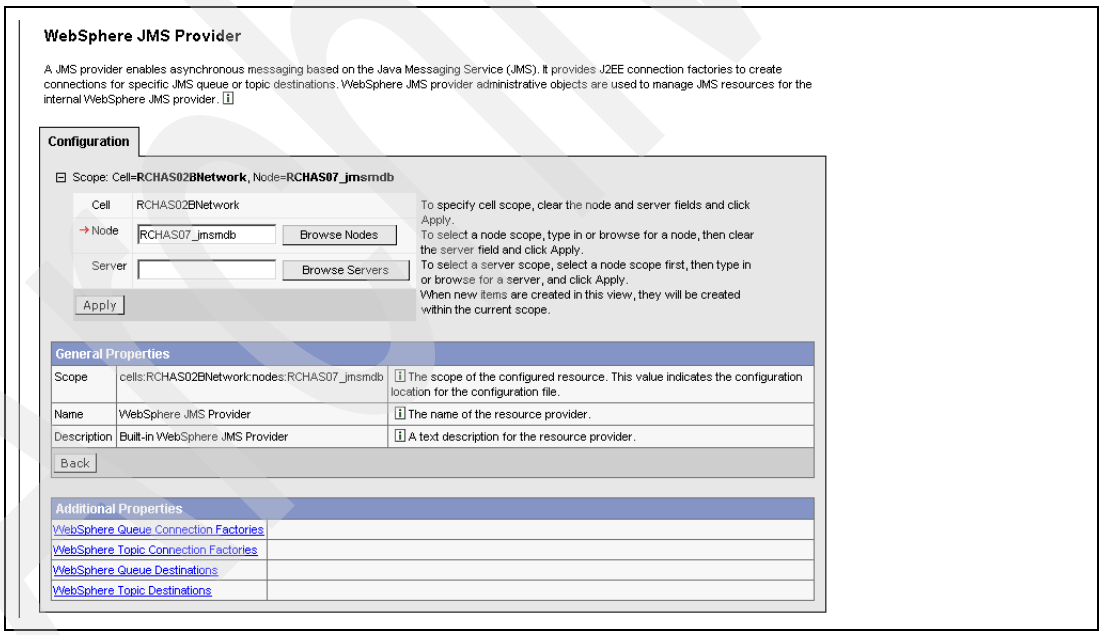


Figure 7-58 Clear selected server and use apply, that changed to node scope

10. By clicking the appropriate link, we can see what JMS resources are defined at this node. For example, SampleJMSTopicConnectionFactory was created for our jmsmdb sample; see Figure 7-59.

Figure 7-59 SampleJMSTopicConnectionFactory

7.5.2 Starting and stopping a JMS server in the ND environment

In WAS-ND a JMS server runs in its own JVM, so you can start or stop Java Message Service (JMS) servers separately from your application servers. In the base WebSphere Application Server product, the JMS server runs as part of each application server and is started, when the application server is started; see “Verifying if the MQ listener job is started” on page 121. In a Network Deployment configuration, you can run the application server process without starting the JMS server.

After you federate a JMS-enabled application server into a Network Deployment cell, the JMS server is created and can be displayed separately in the ND administrative console; see Figure 7-60. You can start, stop, or change the configuration parameters, like the port numbers for the JMS server, from this panel.

To start or stop the JMS server, follow these steps:

1. Start the ND administrative console.
2. In the topology tree, expand **Servers** and click **JMS Servers**. On the JMS Servers collection page, the available JMS server are listed and the current status is shown; see Figure 7-60.
3. Select a **jmsserver** server that you want to start/stop and click the appropriate **Start or Stop** button.

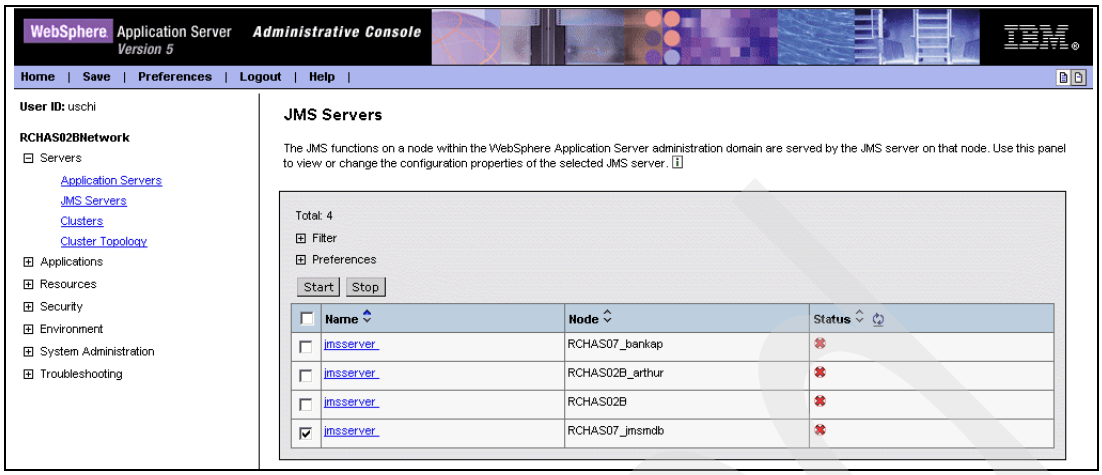


Figure 7-60 Work with JMS Servers

4. You will then see a panel similar to Figure 7-61 with a started JMS server.

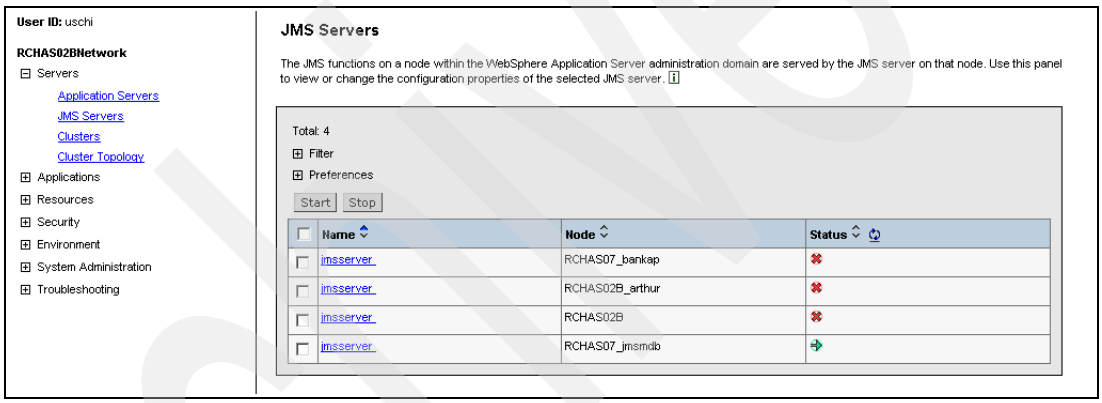


Figure 7-61 jmsserver for RCHAS07_jmsmdb started

5. *QEJBMQLSR*, the MQ listener job, and also the *JMSSERVER* (see "JMS Server job" on page 82), the JMS server job, are started in the iSeries subsystem QEJBAS5 on the federated node (in our case on RCHAS07 for the instance jmsmdb) when you start the JMS server here; see Figure 7-62. When you stop the JMS server from the WAS-ND administrative console, these iSeries jobs are stopped.

   For every JMS server you start via the ND administrative console, you will find two jobs in the QEJBAS5 subsystem with names: QEJBMQLSR and JMSSERVER. To find out to which server (node) each job belongs to, do the steps described in "Verifying if the MQ listener job is started" on page 121 for both of the jobs.

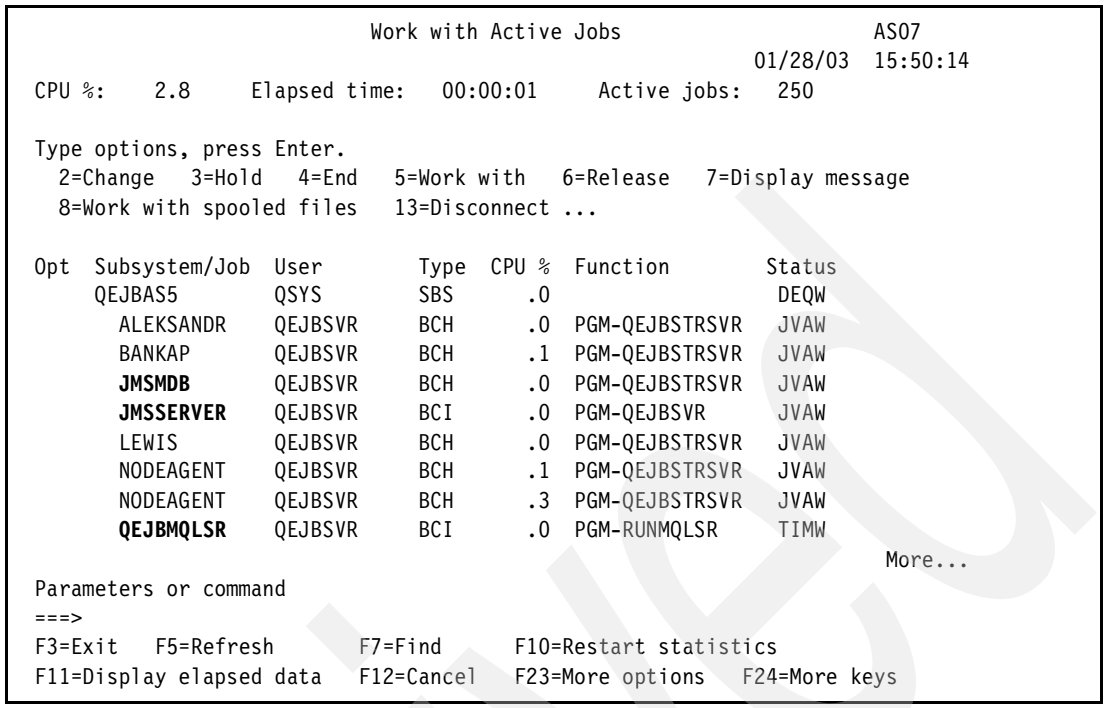


Figure 7-62 JMS server and MQ listener job in subsystem QEJBAS5

7.5.3 Changing the JMS server configuration in the ND environment

The JMS server is created when you federate a node. The available JMS servers can be displayed in the ND administrative console; see Figure 7-60 on page 320. You can start, stop or change the configuration parameters, such as the port numbers for the JMS server, from this panel.

To change the JMS server configuration, follow these steps:

1. Start the ND administrative console.
2. In the topology tree, expand **Servers** and click **JMS Servers**. On the JMS Servers collection page, the available JMS servers are listed and the current status is shown; see Figure 7-60 on page 320.

3. Click the link of the jmsserver you want to change. You will get a panel like the one shown in Figure 7-63, where you have links for the different elements you can change under the Additional Properties section.

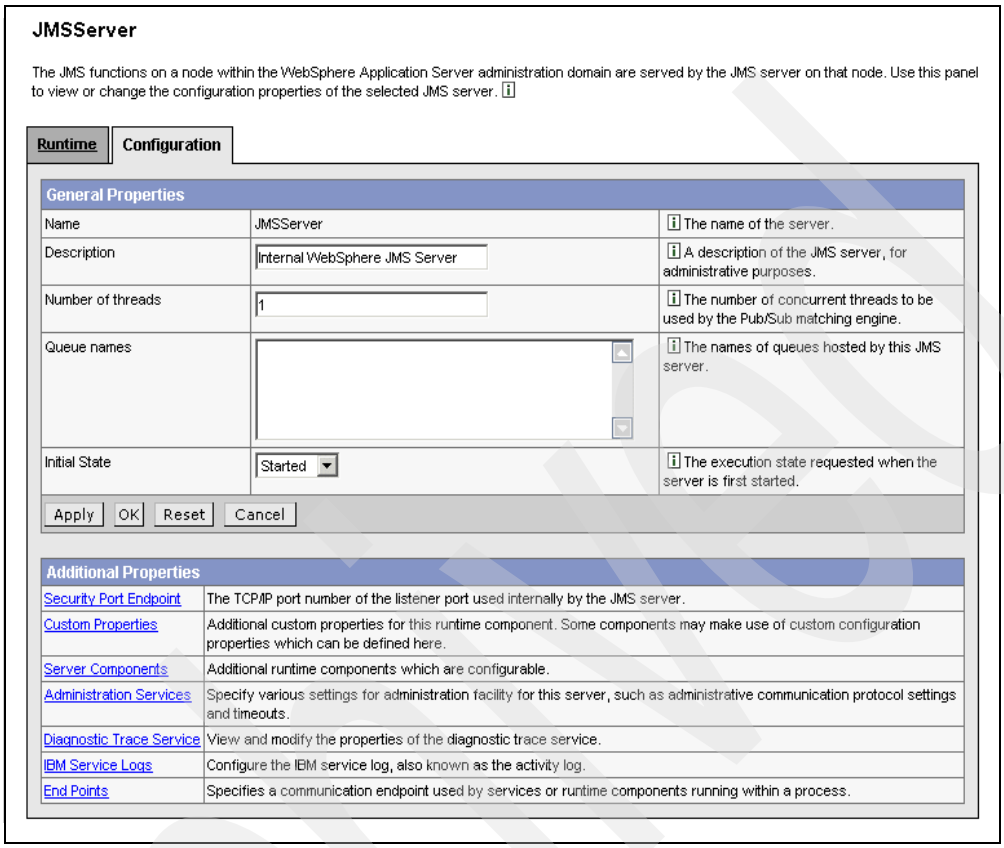


Figure 7-63 JMS server configuration

4. For example, we want to change the value for a TCP/IP port for Bootstrap Address.

   We click **End Points** and get the next panel; see Figure 7-64, where we can choose, via links, one of the four different addresses for services our JMS server uses for communication.

   These are the TCP/IP port numbers for the different services.

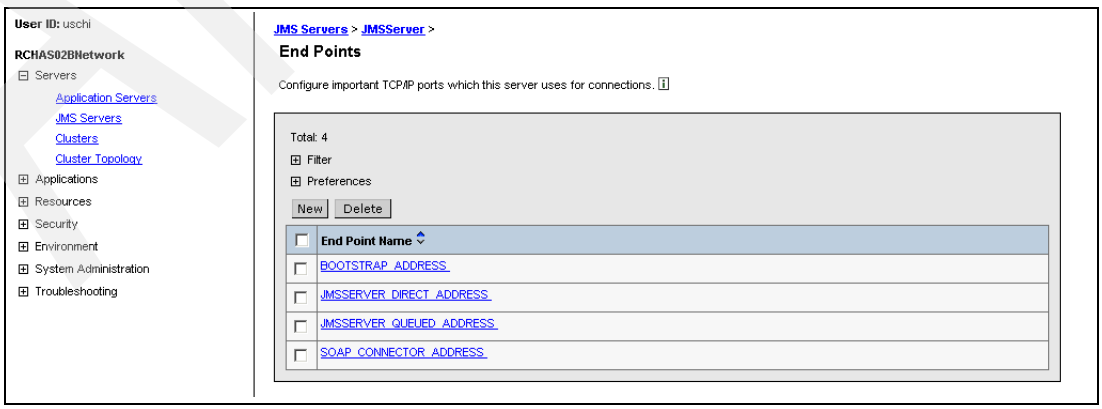


Figure 7-64 End Points for jmsserver

5. Click the link **Bootstrap Address**, and you can see the assigned values; see Figure 7-65.
6. Make your changes here and click **OK**.

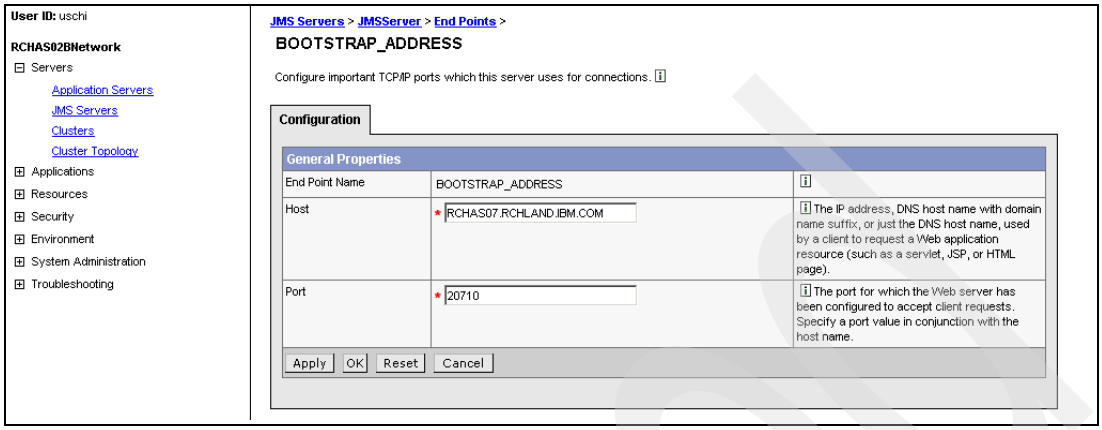


Figure 7-65 Bootstrap Address for jmsserver for node RCHAS07_jmssmd

7. Click **Save** to apply changes to the master configuration.
8. Click the **Save** button on the confirmation page.

7.6 Modifying our JMS sample application

We federated the jmsmdb server on RCHAS07 to the ND cell on RCHAS02B (see 7.3.2, “Adding a node to the network deployment instance” on page 280).

The problem is that our client application is currently bound to the Base Queue Manager for the Embedded JMS Server, WAS_RCHAS07_jmsmdb_jmsmd. The client application binding needs to be changed to WAS_RCHAS07_jmsmdb_jmsserve in order to work in the ND environment.

To modify the bindings for the MDBSamples application, we use the Application Client Resource Configuration Tool. The tool has to be installed on a workstation; see 3.5.3, “Procedure to install workstation tools” on page 64:

1. Start the tool using the `clientConfig.bat` script on a Windows workstation that is found in your %WAS_HOME%\bin directory.
2. Once the tool is running, open the ear file that contains your application. Drill down to your QueueConnectionFactory.
3. Right-click it, and select **Properties** (see Figure 7-66).

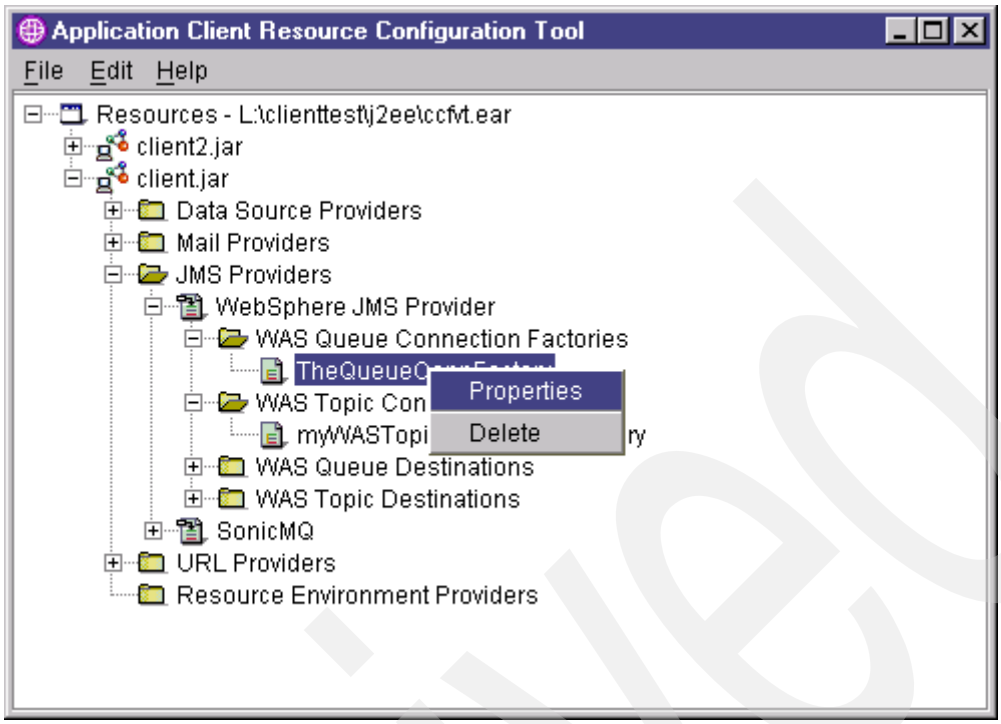


Figure 7-66 Application Client Resource Config Tool - open client.jar

4. Be sure that the Node parameter is set to your node_instance_name. For our sample, it should be WAS_RCHAS07_jmsmd.
5. Modify your Application Server name to `jmsserver`. This will result in the application binding to WAS_RCHAS07_jmsmdb_jmsserver, which, in an ND instance, is the desired behavior.
6. Once these changes are made, click **OK** and save your archive. Your application can now be re-deployed and will bind to the new queue manager name.
7. Start the JMS server; see 7.5.2, “Starting and stopping a JMS server in the ND environment” on page 319.

7.7 Cluster support

Clusters allow you to manage multiple application servers simultaneously and enable workload management.

A cluster is a logical collection of application server processes. You can use clusters to group servers for workload balancing. Application servers within a cluster are called cluster members. All cluster members in a single cluster have identical application components deployed on them. Other than the applications configured to run on them, cluster members do not have to share any configuration data. One cluster member can run on a large multi-processor enterprise server system, while another member of the same cluster can run on a small laptop.

You can use clustering to create identical copies of an application server. A cluster is created based on an existing application server. You can add additional cluster members as needed. It is recommended that you convert the original application server into a cluster member. When you create a new cluster member, it is identical to the original application server.

Clustering applications servers preserves containment relationships. For example, you create a cluster from an application server, which hosts an enterprise application. All cluster members that you create from that cluster also contain the enterprise application.

To manage clustered application servers efficiently, you manage the cluster that contains them. When you make changes to the cluster, those changes are automatically propagated to the cluster members. In addition to making it easy to administer several servers as one logical server, clustering ensures that the cluster members are identical, so that requests are processed in the same manner, regardless of which application server processes the request.

Workload management distributes client requests among application servers in a cluster based on server-weighting. WebSphere Application Server can respond to an increased workload by automatically routing client requests to other nodes as needed.

There are two typical topologies when you use clustering:

- Vertical clusters have servers on the same node under workload management.
- Horizontal clusters have servers on multiple nodes under workload management.

Workload management is enabled when you start the application server cluster. No additional configuration is necessary.

You can use the Deployment Manager administrative console to define clusters. Clusters are analogous to WebSphere Application Server Version 4 server groups. A cell can have no clusters, one cluster, or multiple clusters.

Note: The ability to run different WebSphere Application Server versions on one server does not allow you to include Version 5 application servers in an existing 3.5 or 4.0 administrative domain, or to include Version 3.5.x or Version 4 application servers in a version 5 cell.

On some server platforms, clustering is more important than on others. For example, on NT, it can be critical to spread enterprise workload over several different boxes due to the load capability of a NT server. With iSeries we have lots of horsepower, and the need for horizontal clustering is diminished from a scalability standpoint.

Let us now consider high availability (HA), which really has two important components. First, protection from any kind of outage is a requirement. Second — and this is key — you need the ability to upgrade an application, systems release level, WAS level, etc., without bringing down your business. This upgrade ability is difficult to obtain with clustering alone. Instead, we have gone with applications deployed in two separate instances and then used a Network Dispatcher up-front to distribute the requests to two HTTP servers. This also provides HTTP failover, which is difficult to achieve with just clustering alone. You can find some good information about HA for iSeries at:

http://publib.boulder.ibm.com/iseries/v5r2/ic2924/index.htm?info/experience/whatopabstract

7.7.1 Vertical scaling sample topology

Vertical scaling refers to the topology that exists when you create a cluster of the application servers on a single physical system. This topology requires WebSphere Application Server Network Deployment. Figure 7-67 shows an example of a vertical scaling topology.

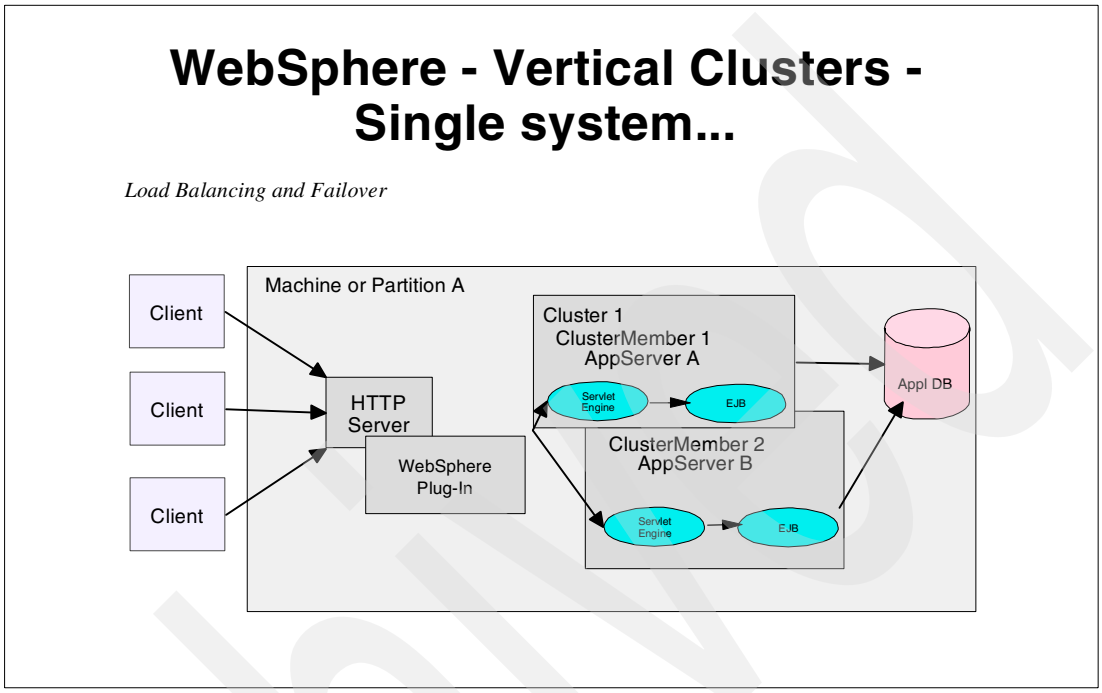


Figure 7-67 Vertical scaling

In this sample topology, multiple cluster members of an application server work together to implement vertical scaling on Machine A. You can also combine vertical scaling with other high availability techniques and configurations to improve availability, performance, and throughput.

Vertical scaling offers the following advantages:

- **Processing power:** Potentially more efficient use of the machine's processing power is possible. Each application server runs in a single Java virtual machine (JVM) process. Some applications may create contention points, or bottlenecks, on resources scooped to the JVM. For some applications, the use of multiple clustered servers, each with its own JVM process, improves availability by reducing the overall contention.
- **Multiple servers:** Vertical scaling provides a straightforward mechanism for creating multiple servers of an application server (and multiple JVM processes) on a single physical machine. This enables the application server to make the best possible use of the processing power of the host machine.
- **Load balancing:** Vertically clustered application servers can use workload management to improve availability, performance, and throughput.
- **Process failover:** A vertical scaling topology provides failover support among application server cluster members. If one application server fails, the other application server cluster members on the machine continue to process client requests.

The primary disadvantage of single-machine vertical scaling topologies is that the host machine becomes a single point of failure in the system. To eliminate this risk, you can use horizontal scaling.

Recommendations for configuration a vertical scaling topology

Before you implement vertical scaling, you must decide how many cluster members to create. These are some factors to consider as you plan for a vertical scaling topology:

- **Application design:** Applications that use more components require more memory, which limits the number of cluster members that you can run on a machine.
- **Hardware environment:** Vertical scaling is most beneficial on machines with large amounts of memory and processing power. However, eventually the overhead of running more cluster members cancels out the benefits of adding them.

The best way to ensure good performance in a vertical scaling configuration is to tune a single application server for throughput and performance, then incrementally add cluster members if needed. In general, a single application server on iSeries is adequate to process most workloads. Test performance and throughput as you add each cluster member. Always monitor memory use when you are configuring a vertical scaling topology to ensure additional memory requirements do not cause excessive paging on a machine.

7.7.2 Horizontal scaling sample topology

Horizontal scaling refers to setting up multiple application server cluster members on two or more physical machines or logical partitions within a single WebSphere Application Server cell. In a horizontal scaling topology, an application can run on all of the machines and present a single system image to clients. This topology requires WebSphere Application Server Network Deployment.

In horizontal scaling, all of the machines host a cluster member for each application server cluster. For example, suppose you have two machines (Machine1 and Machine2) and two application server clusters (Cluster1 and Cluster2). Each cluster contains two cluster members, and each machine hosts one of the cluster members from each cluster.

- Cluster1 includes cluster members named ClusterMember1 and ClusterMember2.
- Cluster2 includes cluster members named ClusterMember3 and ClusterMember4.
- Machine1 hosts ClusterMember1 and ClusterMember3.
- Machine2 hosts ClusterMember2 and ClusterMember4.

Figure 7-68 shows an example of this horizontal scaling topology.

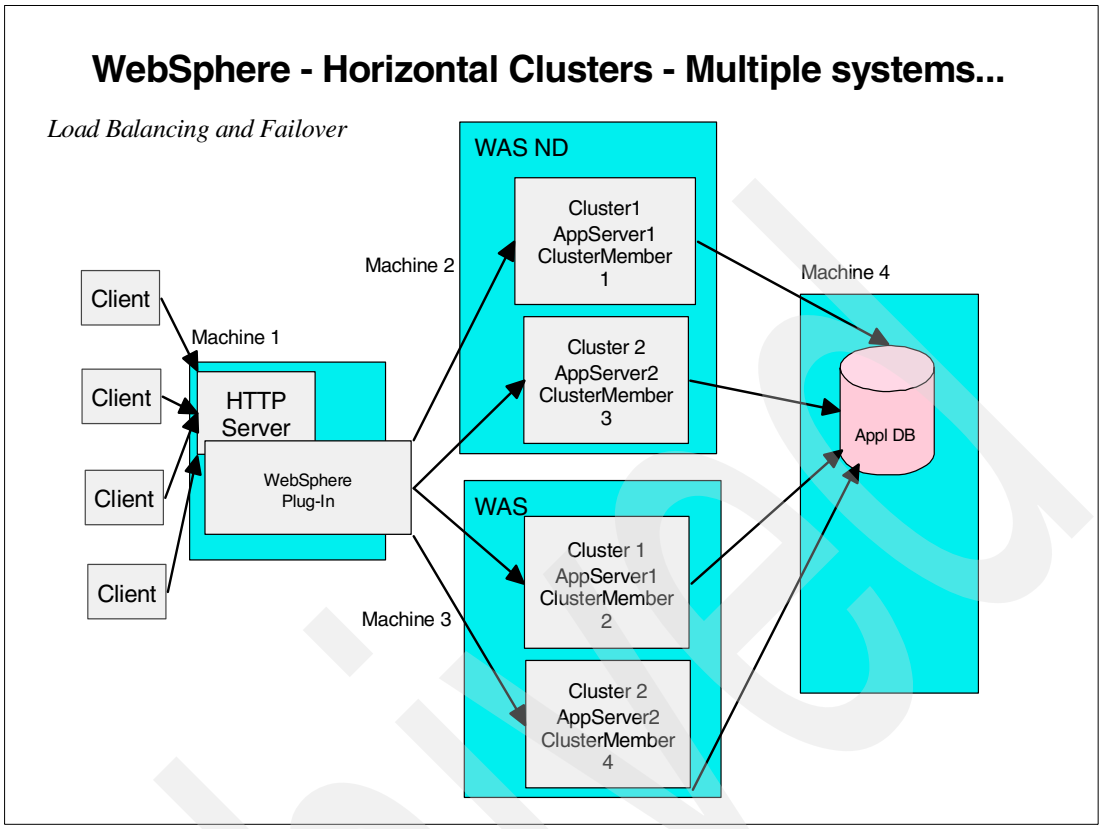


Figure 7-68 Horizontal scaling

Figure 7-69 shows the components that are involved in more detail. This sample topology includes these features:

- The use of two iSeries servers and LPAR provides process and data isolation, as well as hardware redundancy.
- Both Machine A and Machine B host a cluster member of each application server cluster.
- A partition on Machine A serves as the primary Web server for the application and distributes client requests to the application server cluster members on both machines. With the Web server in a separate partition, you can set up a firewall between the Web server and your application server. If you do not need the additional security of a firewall, it is not necessary to run the Web server in a separate partition.
- A single Network Deployment manager communicates with a node agent on each machine to manage the application servers.

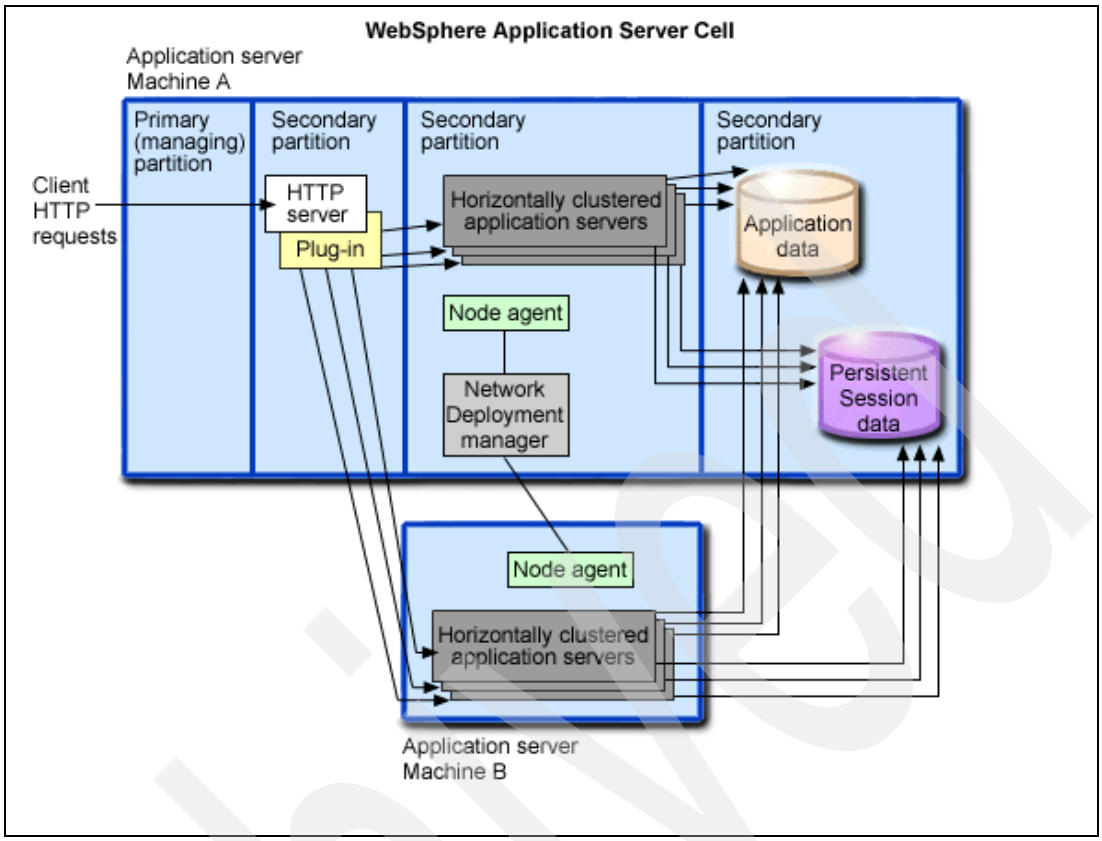


Figure 7-69 Horizontal Clustering

You can include additional components to create more advanced horizontal scaling topologies. Two types of variations on the horizontal clustering topology are listed here:

- Horizontal scaling with Network Dispatcher
- Horizontal scaling with high availability Apache Web server

For more information about variations of horizontal clustering see the WAS online documentation at:

`http://publib.boulder.ibm.com/iseries/v1r1m0/websphere/ic2924/index.htm?info/rzaiz/50/was`

Expand **Administration -> Advanced topologies**.

Horizontal scaling provides these advantages:

- Increased throughput and failover support when compared to vertical scaling topologies.
- Improved availability, because application server process failover and hardware failover support prevent significant interruptions to client service.
- Ability to optimize the distribution of client requests through mechanisms such as workload management or remote HTTP.

The primary disadvantage of a horizontal scaling topology is that it is more complex to administer and maintain than a vertical scaling topology or a single machine topology.

7.7.3 Process of creating a cluster

The basic process of creating a cluster consists of these steps:

1. Before creating a cluster, configure the application server that you want to use to create it.
2. Install the applications that you want to run in your cluster.
3. Configure resources for the applications and the application server.
4. Verify and test the application.
5. Use the ND administrative console to create a cluster based on your application server, and include the original application server as a cluster member in the new cluster.
6. Create additional cluster members based on the original application server.
7. With the ND managers synchronization feature, application specific files will automatically be deposited on nodes within the cluster.

For horizontal clustering, follow this additional advice:

1. Create a WAS instance on both machines (for example, crtwasinst -instance xyz).
2. Federate both nodes (servers) into the ND manager (addNode -instance xyz).
3. Now manage your applications with the ND manager. It is easier to manage everything with ND — so federate as early as possible.
4. Install applications within application servers.
5. Create a cluster.

> **Note:** You could create the cluster first and then install the application

6. Create cluster members.

Creating a cluster

To create a cluster, follow these steps:

1. Start the administrative console; see “Working with the ND administrative console” on page 298.
2. In the topology tree, expand **Servers** and click **Clusters**.
3. On the Server Cluster page, click **New** (see Figure 7-70).

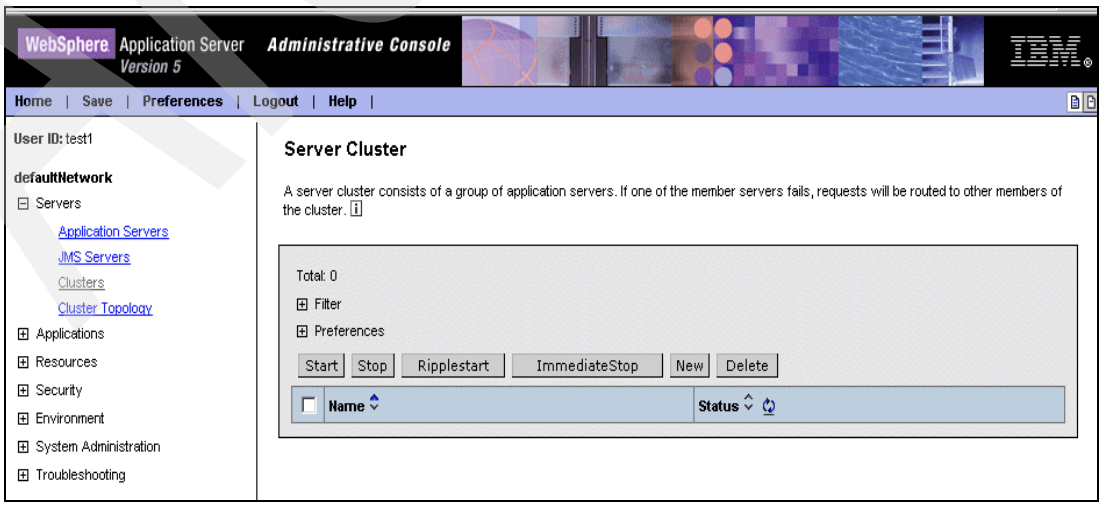


Figure 7-70 New Cluster

4. On the Create New Cluster page, type a name for the cluster.
5. By default, the **Prefer local** checkbox is selected. It specifies whether enterprise bean requests will be routed, when possible, to the node on which the client resides.
6. Select the **Create Replication Domain** for this cluster checkbox to enable memory-to-memory replication of HttpSession (for failover) or replication of cached data and cache invalidations with a Web container's dynamic caching.
7. Choose whether to include an existing application server in the cluster:
   - To create an empty cluster, select Do not include an existing server in this cluster.
   - To create a cluster based on an existing server, select **Select an existing server to add to this cluster** and select the application server from the drop-down list. It is recommended that you include an existing server.
8. Specify the **server weight** (see Figure 7-71.

   The weight value controls the amount of work directed to the application server. If the weight value for the server is greater than the weight values assigned to other servers in the cluster, then the server will receive a larger share of the servers' work load. The value can range from 0 to 100.
9. Click **Next**; see Figure 7-71.

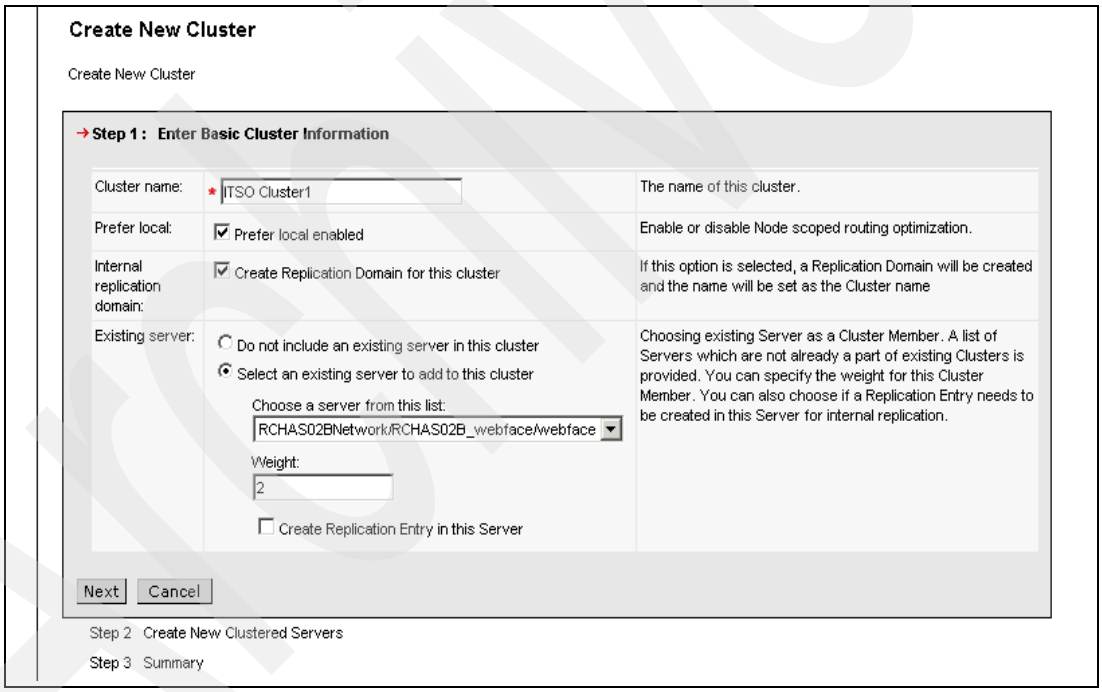


Figure 7-71 Create Cluster Step 1

10. If you want to include more cluster members in the cluster, you can create new application servers (cluster members) to become part of the cluster. For each new cluster member, follow these steps:

> **Note:** This is where vertical clustering is differentiated from horizontal clustering. If you select the same node as the base application server as the first cluster member, then the cluster member is a vertical cluster member. If you select a different node, then it is a horizontal member.

a. Type the **name of a new cluster member** to add to the cluster.

b. Select the **node** on which the server will reside.

c. Specify the **server weight**; see step number 8 on page 331.

d. Specify whether to generate a unique HTTP port.

e. Specify whether to create a replication entry for the server.

f. Click **Apply**.

The additional cluster member is created; see Figure 7-72.

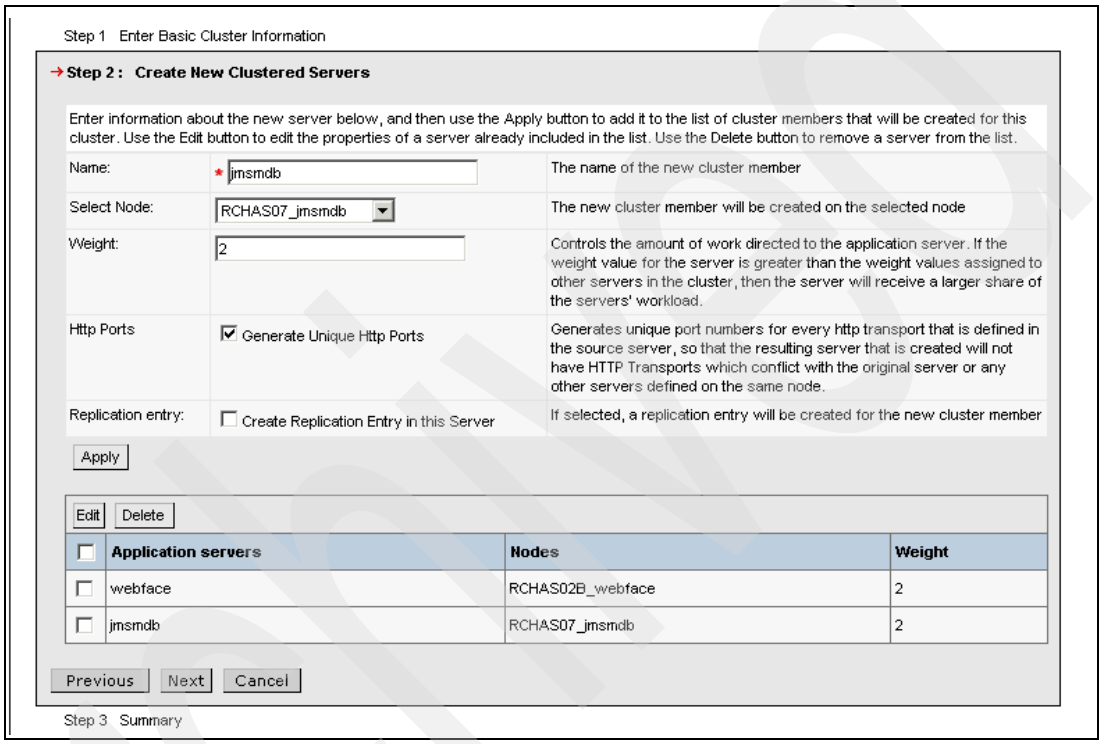


Figure 7-72 Create cluster step 2 - add second cluster member

11. Click **Next.** In the summary panel you can see the information you defined for the new cluster (see Figure 7-73).

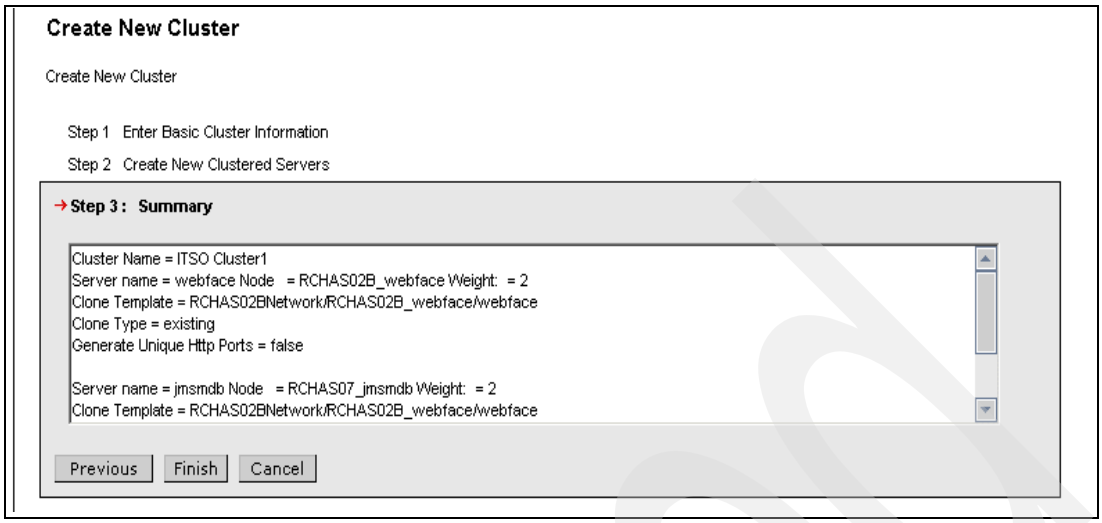


Figure 7-73 summary ITSO cluster 1

12. Click **Finish**.

13. You can expand **View items with changes** to see the configuration files that are updated or added (see Figure 7-74).

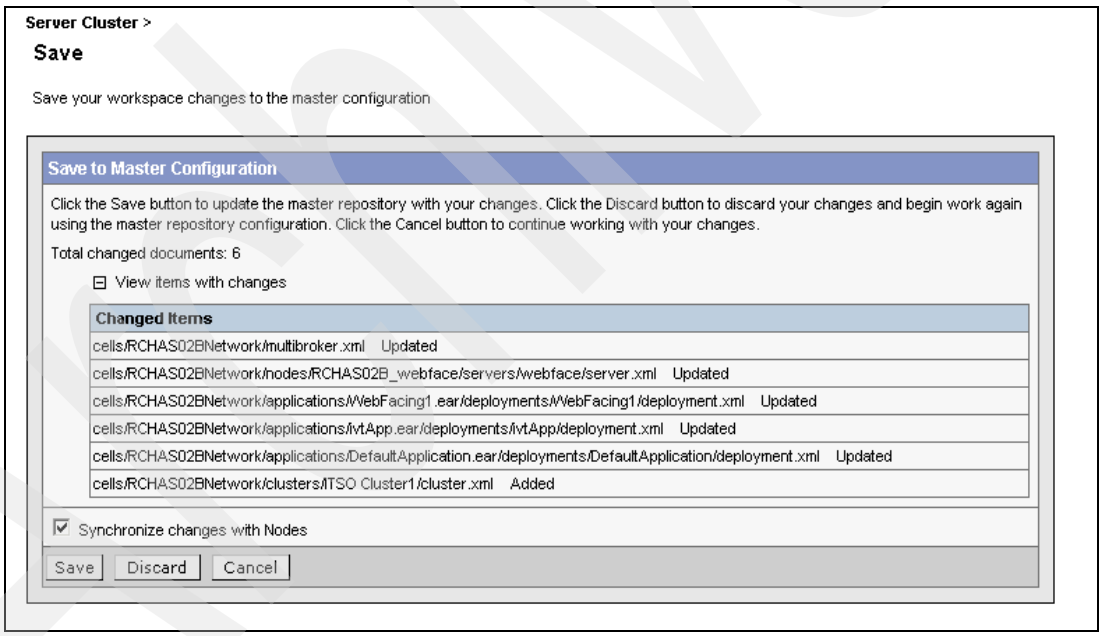


Figure 7-74 Changes to the configuration

14. **Save** the configuration.

After creating clusters and cluster members, you will find new directories in the IFS on your iSeries for the clusters which hold the cluster.xml configuration file, as shown in Figure 7-75.

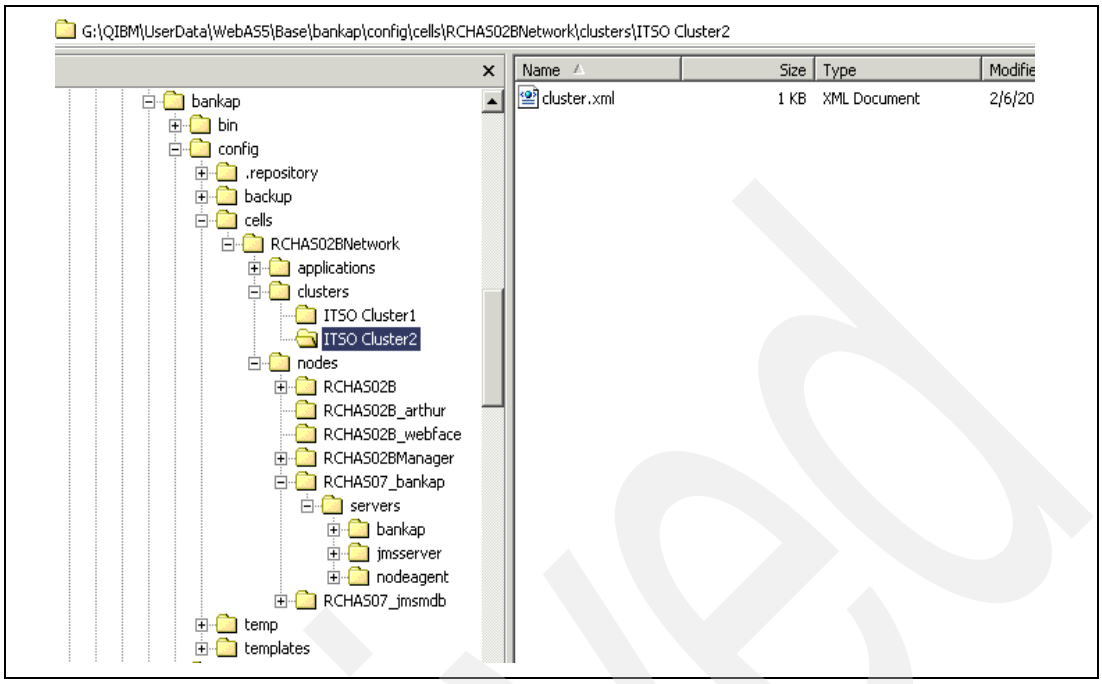


Figure 7-75 IFS directory structure of instance bankap on RCHAS07 iSeries server

7.7.4 Displaying cluster topology

You can use the administrative console to see the topology of your defined clusters in a cell:

1. Start the administrative console; see “Working with the ND administrative console” on page 298.
2. In the topology tree, expand **Servers** and click **Cluster Topology**.
3. Expand the shown clusters and the nodes to see all nodes and cluster members in your cluster; see our sample in Figure 7-76.

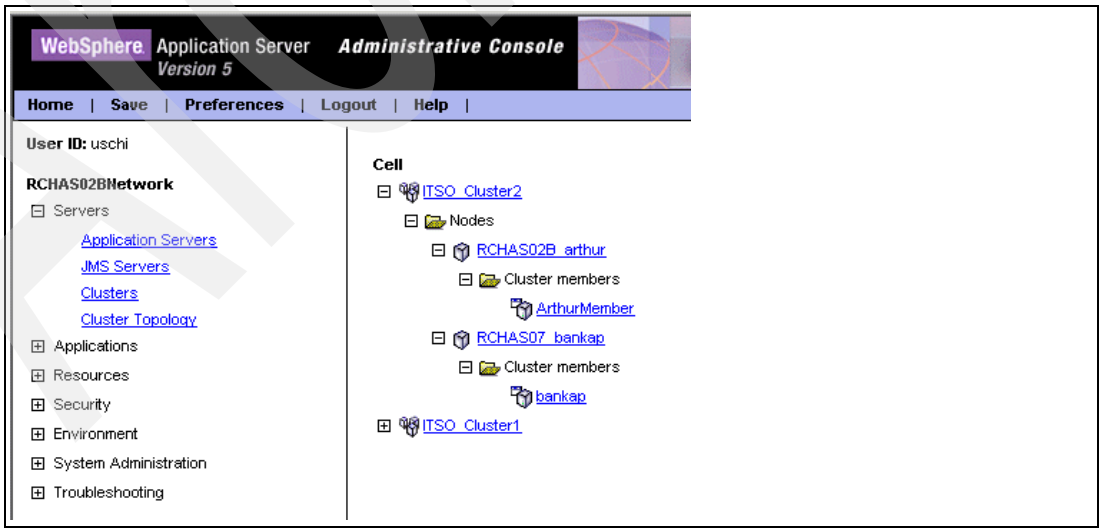


Figure 7-76 Cluster Topology RCHAS02B - Cluster 2

7.7.5 Starting a cluster

When you request that all members of a cluster start, the cluster state changes to Partial Start, and each server that is a member of that cluster launches, if it is not already running.

The Start action launches the server process of each member of the cluster by calling the node agent for each server to start the servers. After all servers are running, the status of the cluster changes to Started. If the call to a node agent for a server fails, the server will not start.

RippleStart combines stopping and starting operations. It first stops and then restarts each member of the cluster.

Note: Only the servers of a node where the node agent is already started can be started in this way.

If the Node Agent is not started, you will get messages similar to those shown in Figure 7-77.

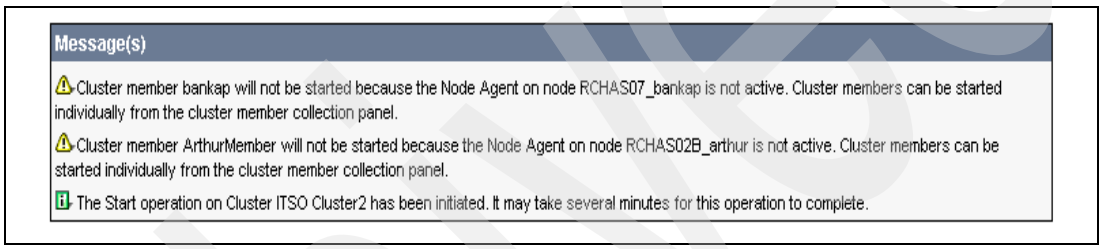


Figure 7-77 Messages by start cluster, when Node Agent is not started.

To start a cluster and the application servers in that cluster, follow these steps:

1. Start the administrative console.
2. In the topology tree, expand **Servers** and click **Clusters**.
3. On the Server Cluster list page, select the cluster whose members you want to start and click the **Start** button (or use the **Ripplestart** button; see Figure 7-78).

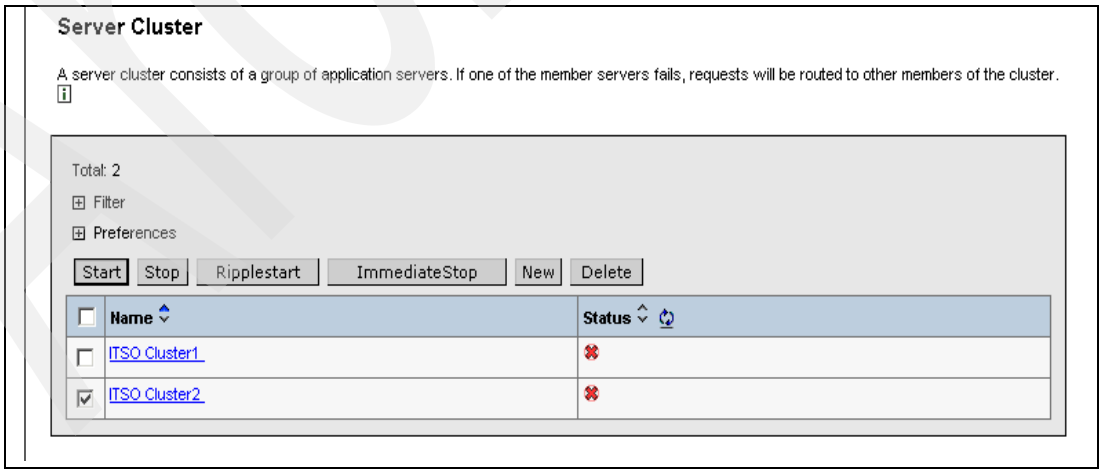


Figure 7-78 Cluster list

7.7.6 Stopping a cluster

Use the administrative console to stop the cluster and all application servers in that cluster:

- *Stop* halts each server in a manner that allows the server to finish existing requests and allows failover to another member of the cluster.
- *Immediate Stop* brings down the server quickly without regard to existing requests.

When the stop operation begins, the cluster status changes to Partial Stop. After all servers stop, the cluster status becomes Stopped.

To stop a cluster and the application servers in that cluster, follow these steps:

1. Start the administrative console.
2. In the topology tree, expand **Servers** and click **Clusters**.
3. On the Server Cluster list page, select the cluster whose members you want to stop and click the **Stop** button (or use the **ImmediateStop** button; see Figure 7-79).

7.7.7 Modifying a cluster

To modify a cluster, follow these steps:

1. Start the administrative console; see “Working with the ND administrative console” on page 298.
2. In the topology tree, expand **Servers** and click **Clusters**. You will get the list of clusters in your cell, as shown in Figure 7-79.

Figure 7-79 Cluster list

3. On the Server Cluster page, click the cluster's name link to view the settings for the cluster. There are tabs showing different information:

   a. **Configuration** tab; see Figure 7-80.

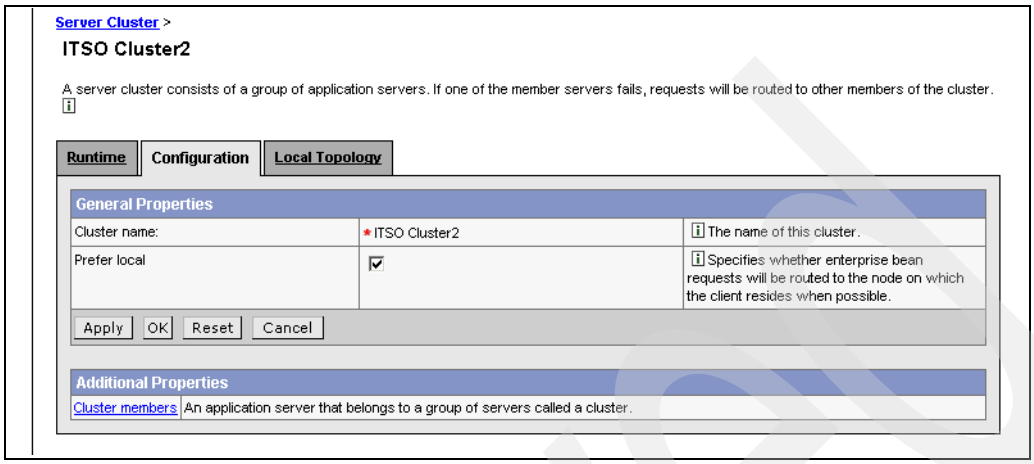


Figure 7-80 Cluster configuration tab

Make your changes for the cluster here.

   b. **Topology** tab for that cluster; see Figure 7-81.

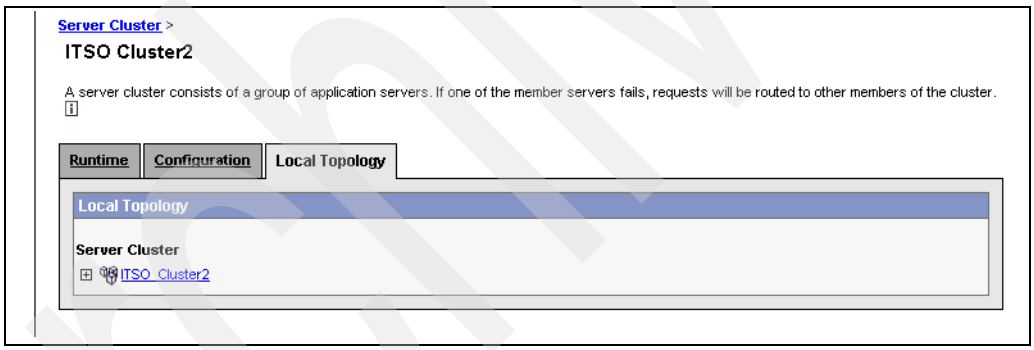


Figure 7-81 Topology tab for that cluster

Expand the cluster here and you will see all the servers (nodes) defined in the cluster; see Figure 7-82.

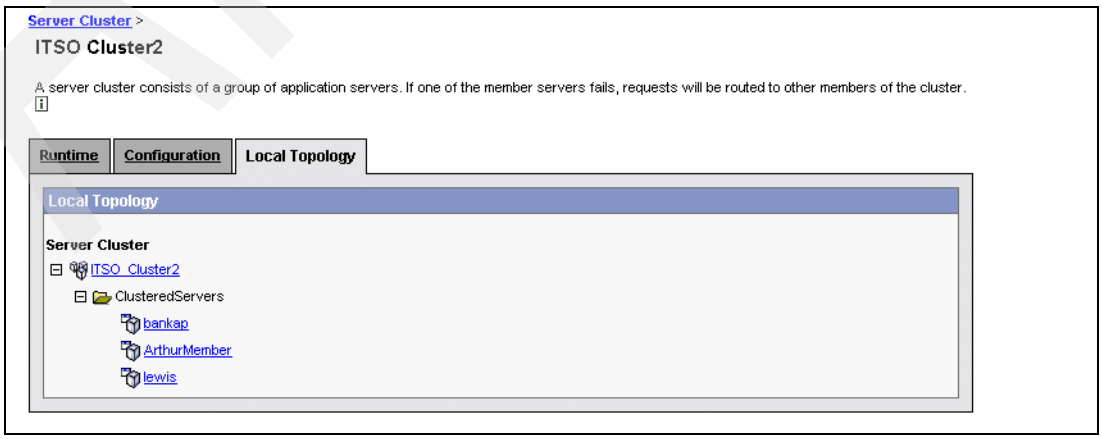


Figure 7-82 Topology tab - expanded with links

You can use the link to get the properties page for the node in the cluster; see Figure 7-83.

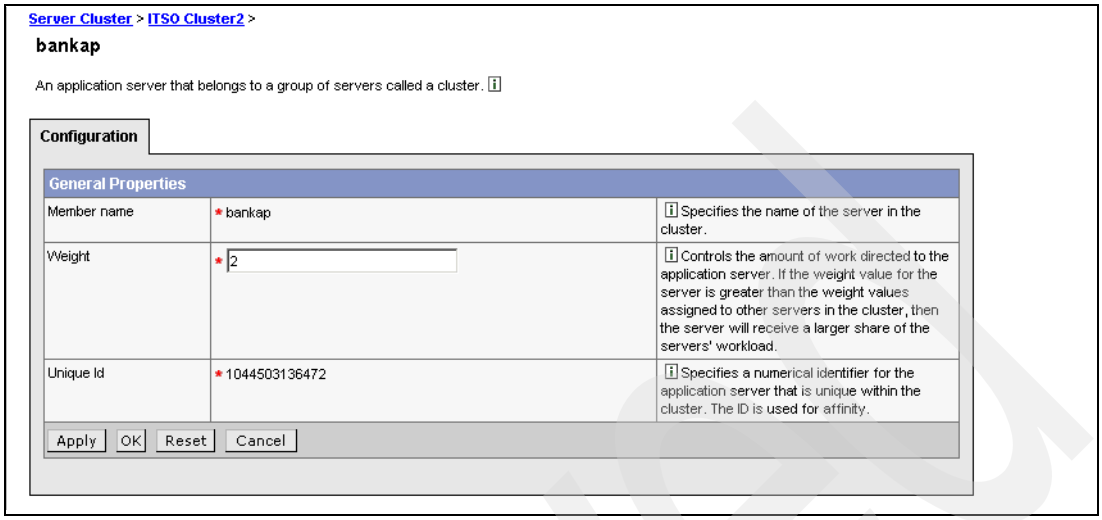


Figure 7-83 Properties for node in a cluster

Make your changes for the Weight properties for the node here.

c. **Runtime** tab; see Figure 7-84.

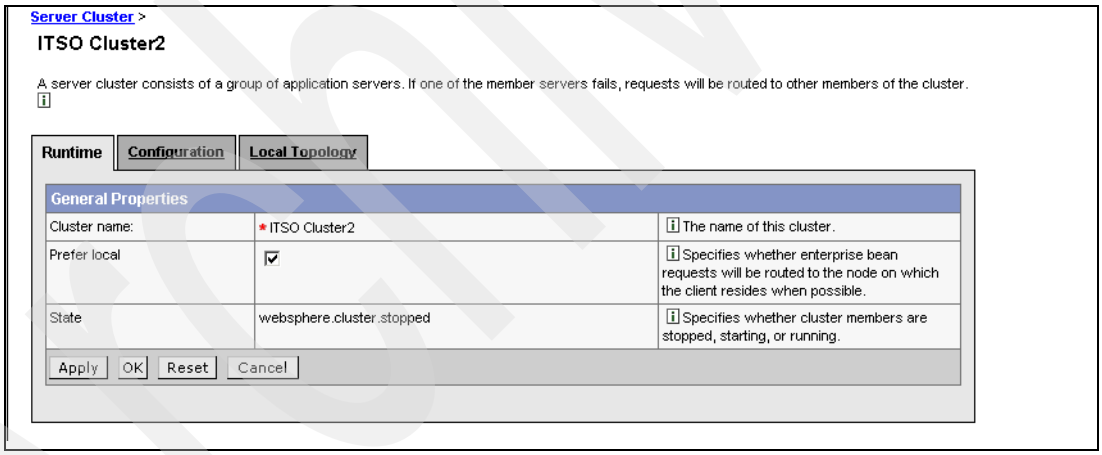


Figure 7-84 Runtime tab

Note: If you do not save the administrative configuration, only the Configuration and Local Topology tabs are available. After you save the administrative configuration, the Runtime tab is also available.

4. Make the changes.
5. Click **OK**.
6. **Save** the configuration.

7.7.8 Removing a cluster

Before you can remove a cluster and all of its cluster members by using the administrative console, you have to uninstall the applications that resides in this cluster. Otherwise you will receive a message similar to the one shown in Figure 7-85.

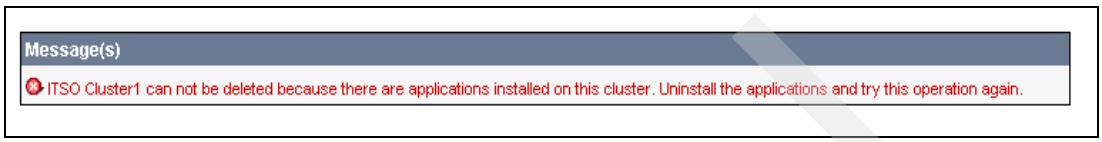


Figure 7-85 Message applications not deleted

To uninstall the applications:

1. Expand the **Applications** in the topology tree of the administrative console.
2. Click **Enterprise Applications**; see Figure 7-86.
3. Select the applications which belong to the cluster you want to remove and click **Stop**.
4. Select the applications which belong to the cluster you want to remove and click **Uninstall** (see Figure 7-86).

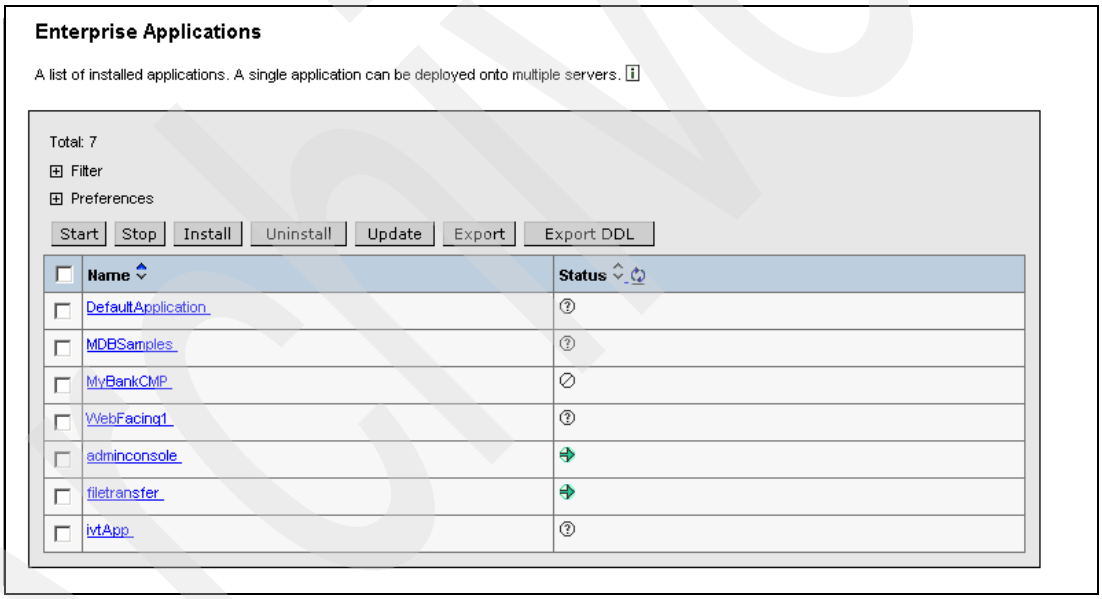


Figure 7-86 Enterprise Applications

To remove a cluster and all of its cluster members, follow these steps:

1. Start the administrative console; see “Working with the ND administrative console” on page 298.
2. In the topology tree, expand **Servers** and click **Clusters**.
3. Select the checkbox for the cluster that you want to remove.
4. Click **Delete**.

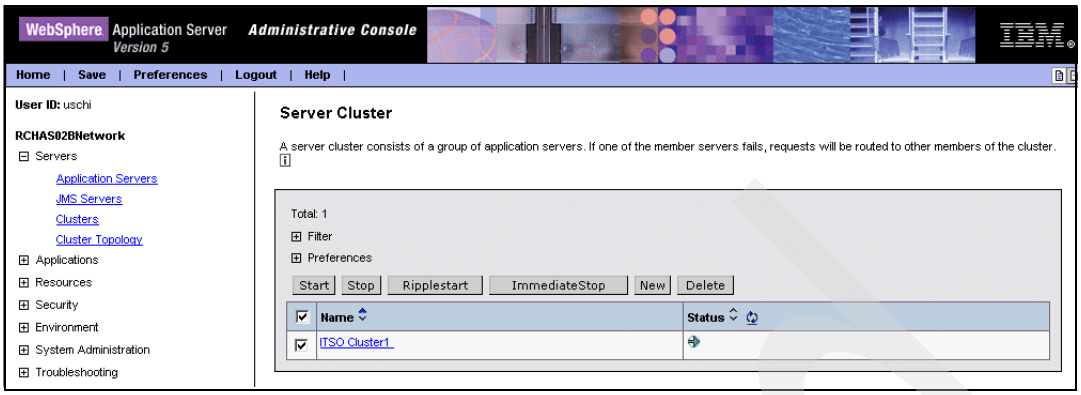


Figure 7-87 Remove cluster

5. **Save** the configuration.

7.7.9 Administer cluster members

In this section we cover the basic administrative tasks for managing the cluster member.

Creating cluster members

As described in “Creating a cluster” on page 330, you can create cluster members in the same step while creating a cluster. Also, it is possible to create cluster members in an existing cluster, which we describe in this section.

To create a cluster member, follow these steps:

1. Start the administrative console.
2. In the topology tree, expand **Servers** and click **Clusters**.
3. On the Server Cluster list page, click the name of the cluster in which you want to create a cluster member.
4. On the Server Cluster page, select the **Configuration** tab and click **Cluster Members**; see Figure 7-88.

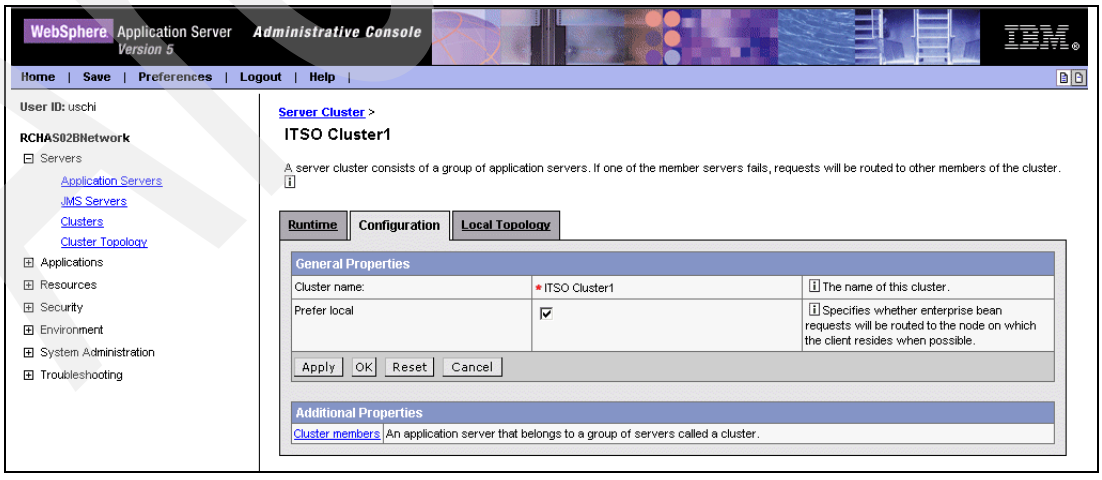


Figure 7-88 Cluster member

5. On the Cluster Members page, click **New** (see Figure 7-89).

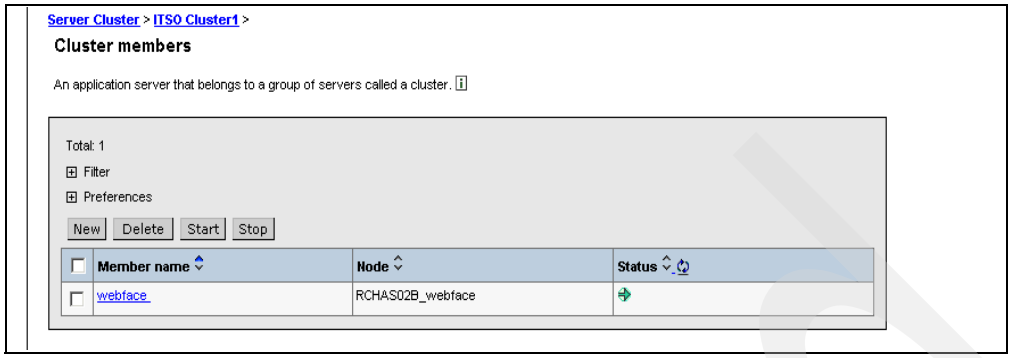


Figure 7-89 Cluster member - new

6. On the Create New Cluster Members page, type a name for the cluster member.
7. Select the node on which you want to create the cluster member.
8. Specify the server weight. The weight value controls the amount of work directed to the application server.
9. Select **Generate Unique HTTP Ports**.
10. Click **Apply**; see Figure 7-90.

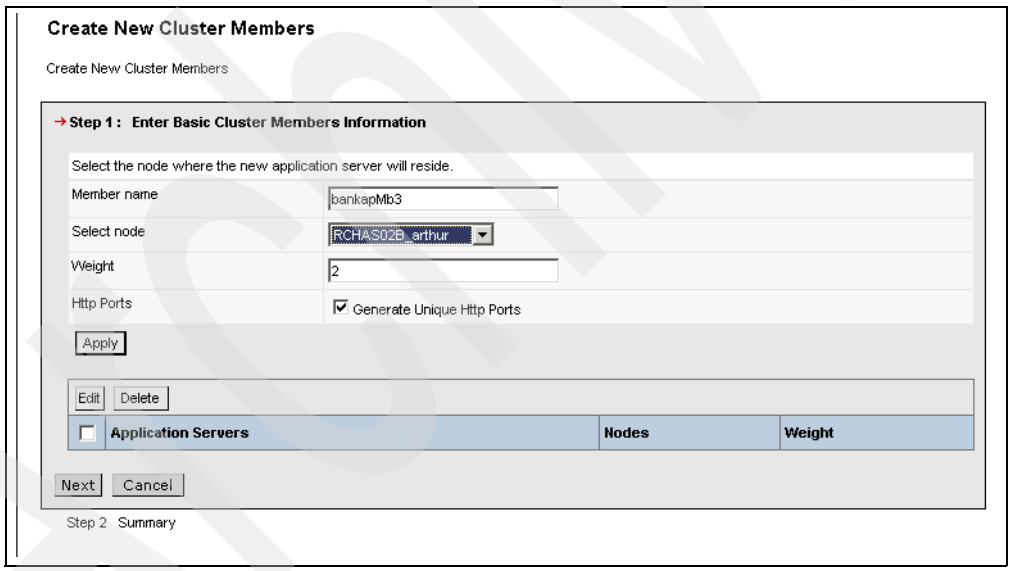


Figure 7-90 Create additional cluster member

11. You will see a panel similar to one shown in Figure 7-91.

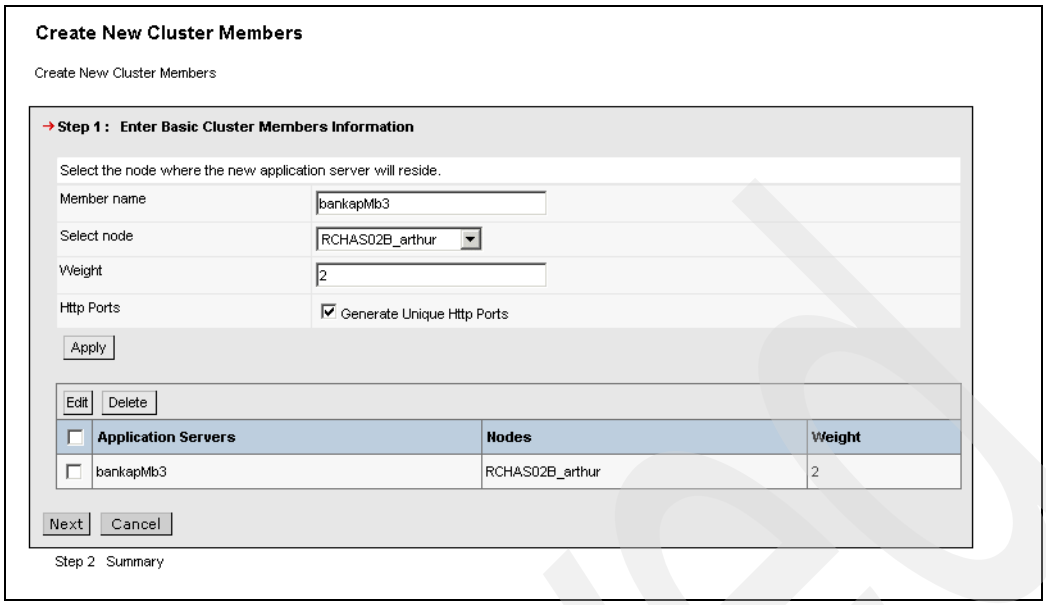


Figure 7-91 Step 1 Cluster member added

12. Click **Next**.
13. Review the information on the Summary page. Notice that, just opposite to the creating of a cluster step, you didn't have to specify a server template (see Figure 7-92).
14. Click **Finish**.

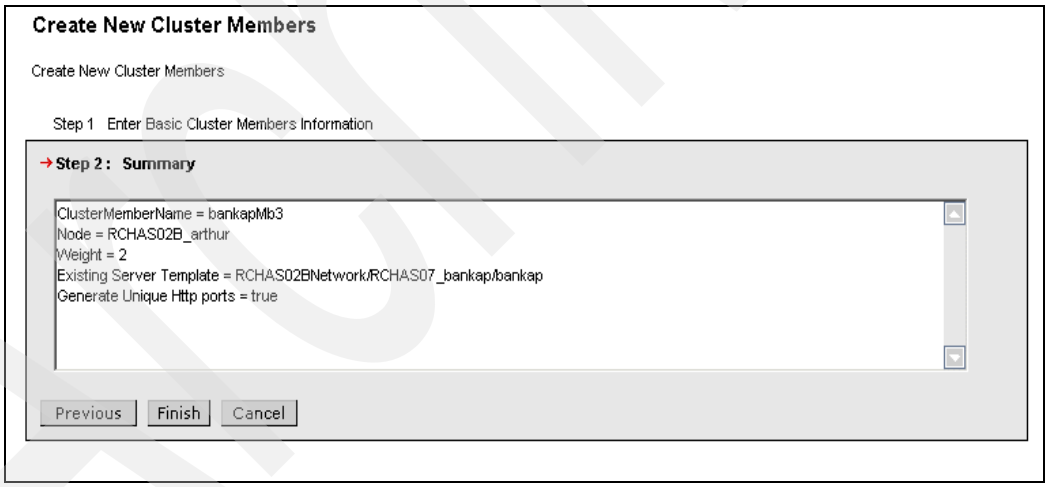


Figure 7-92 Create new cluster member summary list

15. On the first save panel you can expand the **View items with changes**, as shown in Figure 7-93**.**

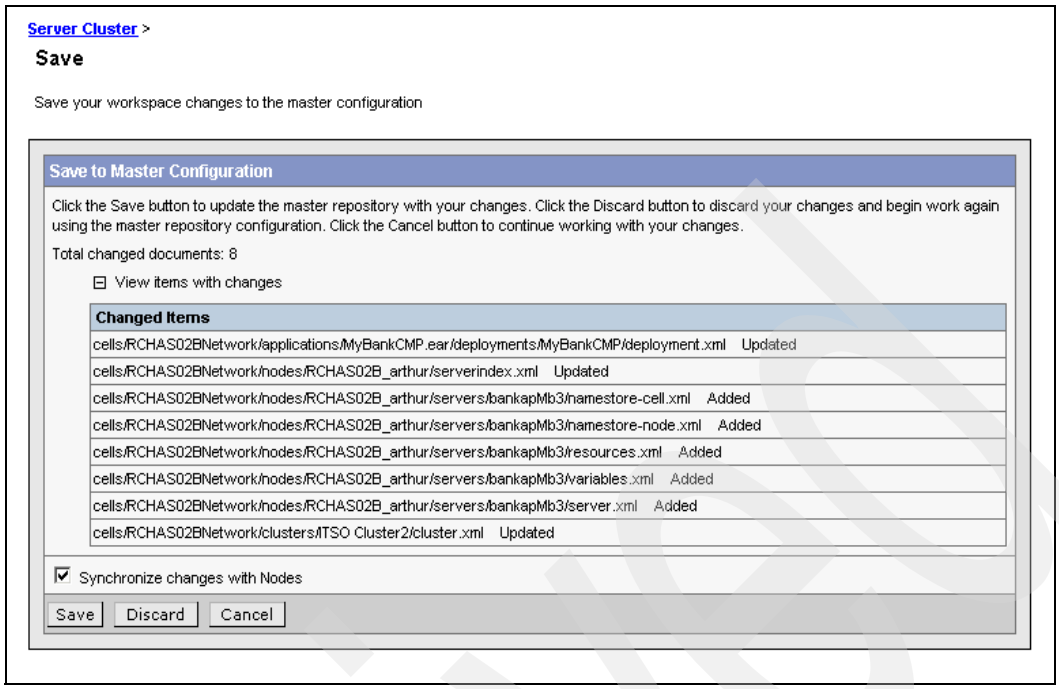


Figure 7-93 Changes to cell configuration

16. **Save** the configuration.

Starting and stopping cluster members

To start or stop a cluster member, or lets say the application server that is a cluster member, you can do the following steps:

1. Start the **administrative console**.
2. In the topology tree, expand **Servers** and click **Clusters**.
3. Click the name of the cluster from which you want to start or stop the cluster member.
4. In the Additional Properties section, click **Cluster Members**.
5. Select the cluster member that you want to start or stop.
6. Click **Start** or **Stop**; see Figure 7-94.

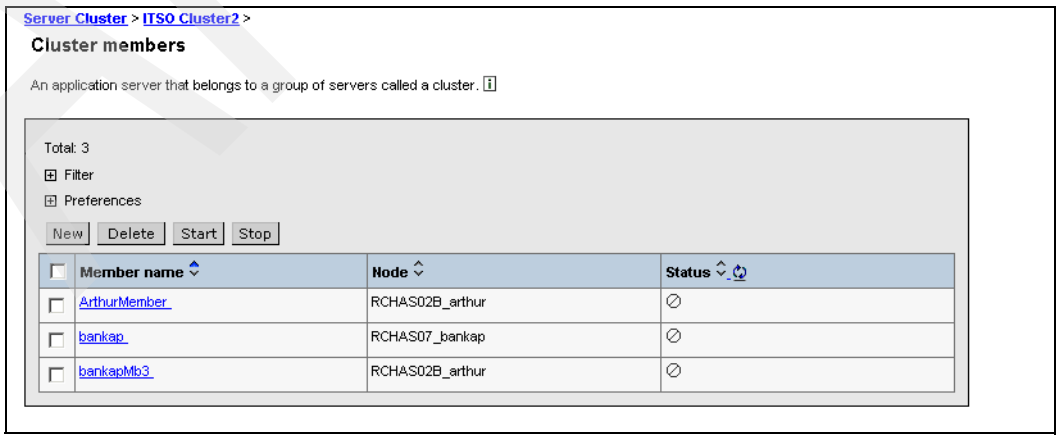


Figure 7-94 Start or Stop cluster member

There is no difference in the behavior if you start or stop an application server in the way described above, or if you use the method **Servers ->Application Servers** as described in 7.4.8, “Starting/stopping an application server via the admin console” on page 314.

The only difference is that, this way, you will see all application servers of a cell as shown in Figure 7-95; and in the way described above, only the application servers of the chosen cluster.

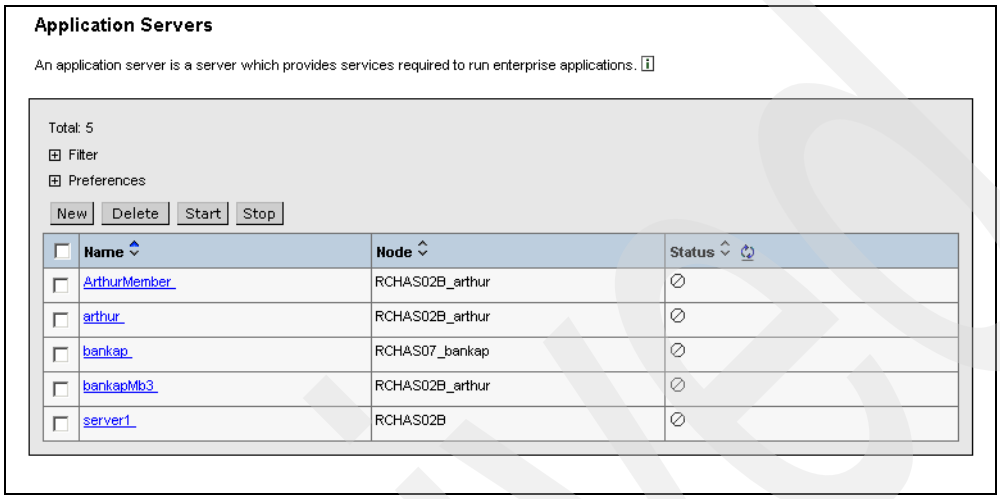


Figure 7-95 Application servers in our cell

Modifying cluster members

To modify a cluster member, follow these steps:

1. Start the administrative console.
2. In the topology tree, expand **Servers** and click **Clusters**.
3. On the Clusters page, click the name of the cluster that contains the cluster member that you want to modify.
4. On the cluster's detail page, click **Cluster Members**.
5. On the Cluster Members page, click the name of the cluster member that you want to modify.
6. Make your changes.
7. Click **OK**.
8. Save the configuration.

Removing cluster members

To remove a cluster member, follow these steps:

1. Start the **administrative console**.
2. In the topology tree, expand **Servers** and click **Clusters**.
3. Click the name of the cluster from which you want to remove the cluster member.
4. In the Additional Properties section, click **Cluster Members**.
5. Select the cluster member that you want to delete (see Figure 7-96).
6. Click **Delete**; see Figure 7-96.

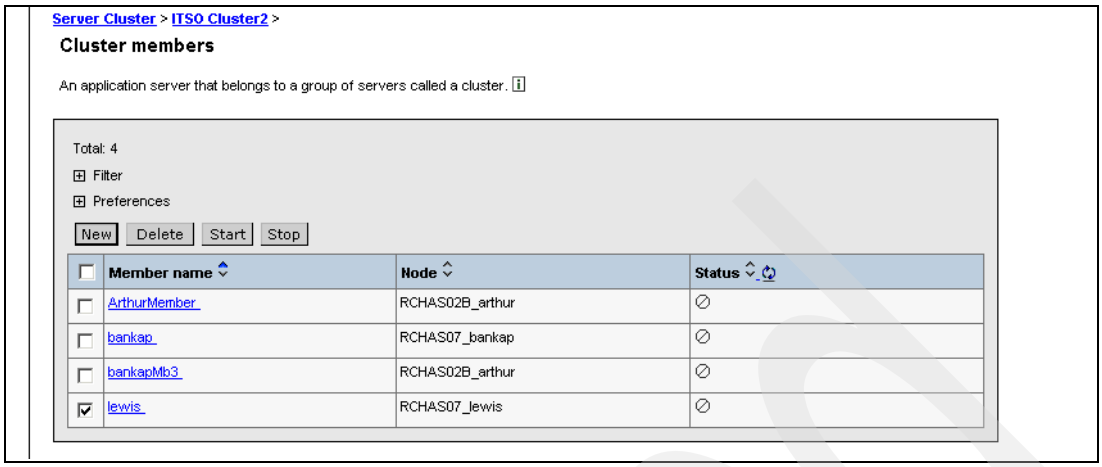


Figure 7-96 remove cluster member

7. **Save** the configuration.

7.7.10 Advice for clustering

Create cluster members based on your knowledge of the application and on the expected workload. Keep the following considerations in mind:

- Cluster members do not need to reside on the same machine.
- Clients can have inconsistent views of configuration information in the cluster. This can occur when an application server is stopped, started, added, or deleted. The period of inconsistency is short-lived, however. Clients eventually refresh their caches of server information. Application servers that are unchanged during the period of inconsistency remain available.
- In most cases, if you make changes to a cluster, you do not need to restart its cluster members and clients. The changes you made eventually propagate to the cluster members and clients. However, in some cases you need to stop and restart the cluster members of the cluster, for example, when you change the selection policy for the cluster.
- You can make changes to a cluster while it is running. However, incremental changes (such as adding or removing one or two cluster members) have less impact on client performance than wholesale changes.
- It is always best to make changes when few clients and application servers are running.
- You can cluster the initial number of application servers based on the expected load for an application. Later, you can respond to the load on an application by adding or removing server cluster members.
- When a machine becomes unavailable, you do not need to reconfigure the cluster members of other application servers to compensate for any unavailable application servers on that machine. However, when the machine is going to be unavailable for an extended period, you might want to reconfigure the other servers to optimize performance.

8

Security

In this chapter we cover the following topics related to WebSphere Application Server security:

- Overview of WebSphere Application Server V5 security
- Setting up global security
- Enabling Secure Sockets Layer (SSL) connections
- Dos and Don’ts while enabling security

8.1 WebSphere Application Server V5.0 for iSeries security

In this section we give you an overview of the various security features. IBM WebSphere Application Server Version 5 (WAS) provides a security infrastructure and mechanisms to protect sensitive J2EE and administrative resources in order to address enterprise end-to-end security requirements on:

- Authentication
- Access control
- Data integrity
- Confidentiality
- Privacy
- Secure interoperability with other J2EE compliant application servers

WAS uses layered architecture to achieve these goals (see Figure 8-1). Its security is based on industry standards. Version 5 has an open architecture that allows secure connectivity and interoperability with Enterprise Information System (EIS) including DB2, CICS®, MQSeries, Domino, IBM Directory, and products from other vendors.

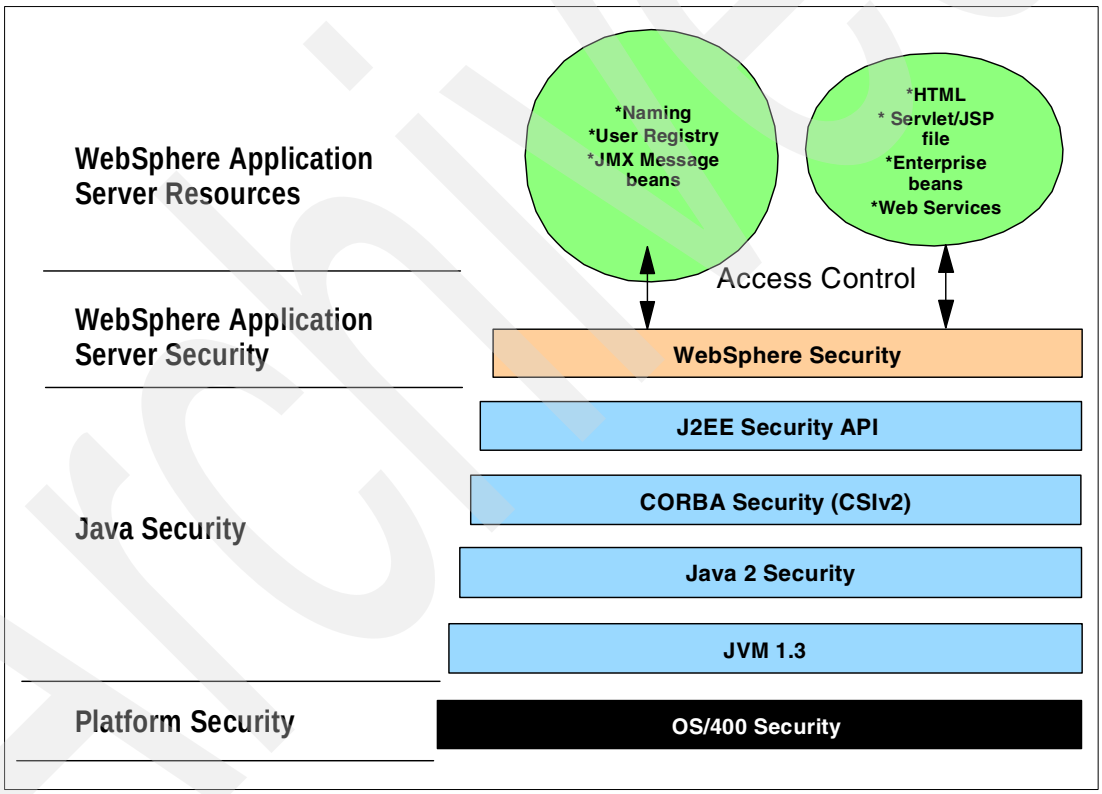


Figure 8-1 Security: layered approach

The product provides a unified, policy-based, and permission-based model for securing Web resources and Enterprise Java according to J2EE specifications. Specifically, Version 5 complies with J2EE specification Version 1.3 and has passed the J2EE Compatibility Test Suite. Product security layered architecture is built on top of OS platform, JVM, and Java 2 security and employs a rich set of security technology, including:

- Java 2 Security model, which provides policy-based, fine-grained, and permission-based access control to system resources.

- Common Security Interoperability Version 2 (CSIv2) security protocol, in addition to the Secure Association Services (SAS) security protocol. SAS protocols is supported by prior product releases. CSIv2 is a new protocol supported by WAS and an integral part of J2EE 1.3 Specification. It is essential for interoperability among application servers from different vendors.
- Java Authentication and Authorization Service (JAAS) programming model for Java applications, Servlets, and EJBs.
- J2EE Connector architecture for plugging in resource adapters that supports access to Enterprise Information Systems.

The standard security model and interfaces supported include Java Secure Socket Extension (JSSE) and Java Cryptographic Extension (JCE) provider for secure socket communication and for message and data encryption.

8.1.1 Open architecture paradigm

An application server plays an integral part of the multiple-tier enterprise computing framework. IBM WebSphere Application Server adopts the open architecture paradigm and provides many plug-in points to integrate with enterprise software components (see Figure 8-2). Plug-in points are based on standard J2EE specifications wherever applicable.

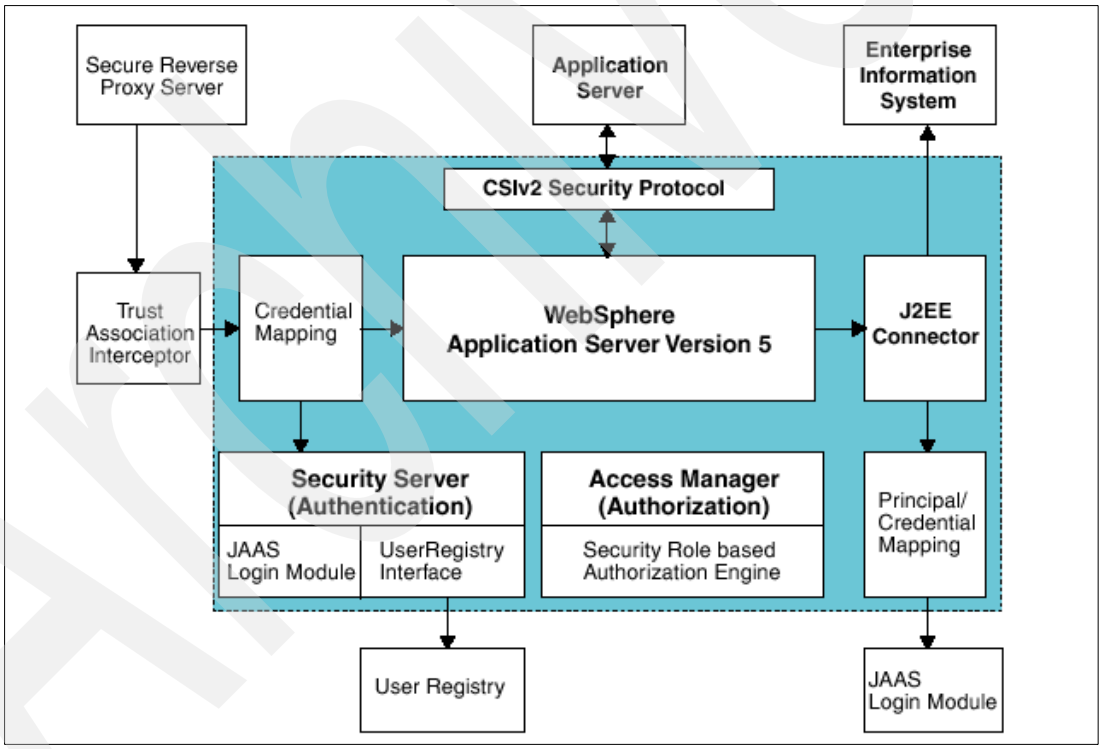


Figure 8-2 WebSphere Application Server security framework

The dotted lines indicate the boundary between the product and other business application components.

The product provides Simple WebSphere Authentication Mechanism (SWAM) and Light Weight Third Party (LTPA) authentication mechanisms (see Figure 8-3). Exactly one of these may be configured to be the active authentication mechanism for the security domain of the product. Exactly one user registry implementation may be configured to be the active user registry of the product security domain.

WAS provides the OS/400 LocalOSUserRegistry and LDAP UserRegistry implementations (see Figure 8-3). It supports a flexible combination of authentication mechanisms and user registries. SWAM is simple to configure and is useful for a single application server environment. LTPA generates a security token for authenticated users which can be propagated to down stream servers and is suitable for distributed environment with multiple application servers. It is possible to use SWAM in a distributed environment if identity assertion is enabled. Note that identity assertion feature is available only on the CSIv2 security protocol.

The LTPA authentication mechanism is designed for distributed security. The security token can be validated by downstream servers. It also supports setting up a trust association relationship with reverse secure proxy servers and Single SignOn (SSO). Besides the combination of LTPA and LDAP or Custom UserRegistry, Version 5 supports LTPA with LocalOSUserRegistry. The new configuration is particularly useful for a single node with multiple application servers.

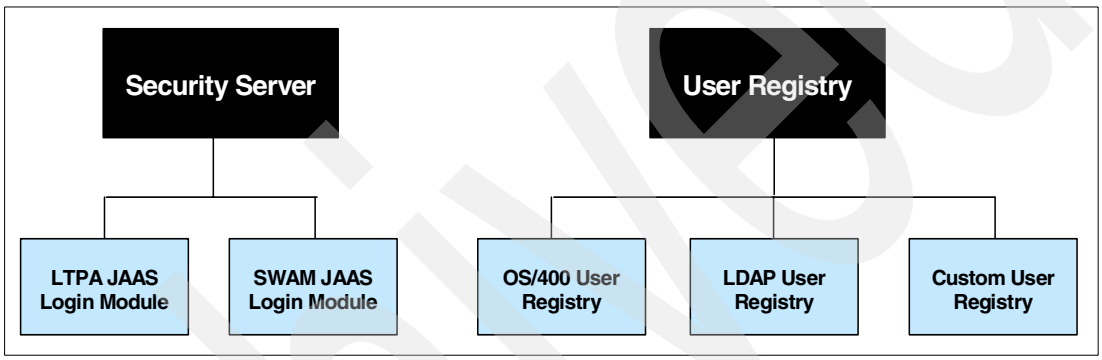


Figure 8-3 Authentication mechanism and user registry

The product supports the J2EE Connector architecture and offers container managed authentication. It provides a default J2C principal/credential mapping module that basically maps any authenticated user credential to a PasswordCredential for the specified EIS security domain (see Figure 8-2 on page 349). The mapping module is a special JAAS LoginModule designed according to the Java 2 Connector and JAAS Specifications. Other mapping LoginModules can be plugged in.

Types of security

Security for J2EE resources is provided by the Web container and EJB container. Each container provides two kinds of security: declarative security and programmatic security. In declarative security, the security structure of an application, including data integrity and confidentiality, authentication requirements, security roles, and access control, is expressed in a form external to the application. In particular, the deployment descriptor is the primary vehicle for declarative security in the J2EE platform. The product maintains J2EE security policy, including information derived from the deployment descriptor and specified by deployers and administrators in a set of XML descriptor files. At runtime, the container uses the security policy defined in the XML descriptor files to enforce data constraints and access control. Those XML configuration files should be protected by operating system security.

When declarative security alone is not sufficient to express the security model of an application, programmatic security may be used by application code to make access decisions. The API for programmatic security consists of two methods of the EJB EJBContext interface (isCallerInRole, getCallerPrincipal) and two methods of the servlet HttpServletRequest interface (isUserInRole, getUserPrincipal).

8.1.2 WebSphere Application Server security architecture

From the perspective of security, every application server process consists of:

- A Web container
- An EJB container
- The administrative subsystem

The security services consists of:

- Authentication mechanism
- User registry
- Access control manager

The product supports the Java 2 Security model. The administrative subsystem, the Web container, and the EJB container code (see Figure 8-4) are running in the product security domain which in the present implementation are granted with AllPermission and can access all system resources. Application code is running in the application security domain which by default is granted with permissions according to J2EE specifications and can only access a restricted set of system resources. The product runtime classes are protected by the product classloader and are kept invisible to application code.

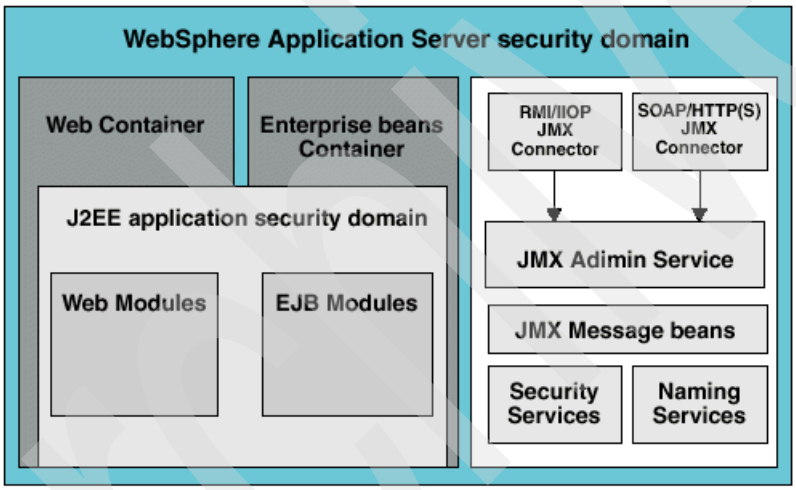


Figure 8-4 WebSphere Application Server security architecture

All of the application server processes by default share a common security configuration that is defined at a cell level security XML document. The security configuration determines whether product security is enforced, whether Java 2 Security is enforced, the authentication mechanism and user registry configuration, security protocol configurations, JAAS login configurations, and Secure Socket Layer configurations.

Applications may have their unique security requirements. Each application server process may create a per-server security configuration to address its own security requirements. Not all security configurations can be modified at the application server level. Those that can be modified at the application server level will include such things as whether application security should be enforced, whether Java 2 Security should be enforced, and security protocol configurations. The administrative subsystem security configuration is always determined by the cell level security document. The Web container and EJB container security configuration are determined both by the optional per server level security document and by the cell level security document, while the former, if it exists, has precedence over the latter.

Security configuration, both at cell level and at application server level, are managed either by the Web based administrative console application or by the wsadmin scripting application.

Web security

When security policy is specified for a Web resource and IBM WebSphere Application Server security is enforced, the Web container performs access control when the resource is requested by a Web client. The Web container will:

- Challenge the Web client for authentication data if none are present, according to the specified authentication method
- Ensure that the data constraints are met
- Determine whether the authenticated user has the required security role

The product supports the following login methods:

- HTTP Basic Authentication
- HTTPS (SSL) Client Authentication
- Form Based Login

Mapping a client certificate to the product security credential uses the UserRegistry implementation to perform the mapping (see Figure 8-5). The LDAP UserRegistry supports the mapping function, while LocalOSUserRegistry does not.

When LTPA authentication mechanism is configured and Single SignOn (SSO) is enabled, an authenticated client will be issued a security cookie which can be used to represent the user within the specified security domain. It is recommended to use SSL to protect the security cookie form being intercepted and from being replayed. When Trust Association is configured, the product can map an authenticated user identity to security credentials based on the trust relationship established with the secure reverse proxy server.

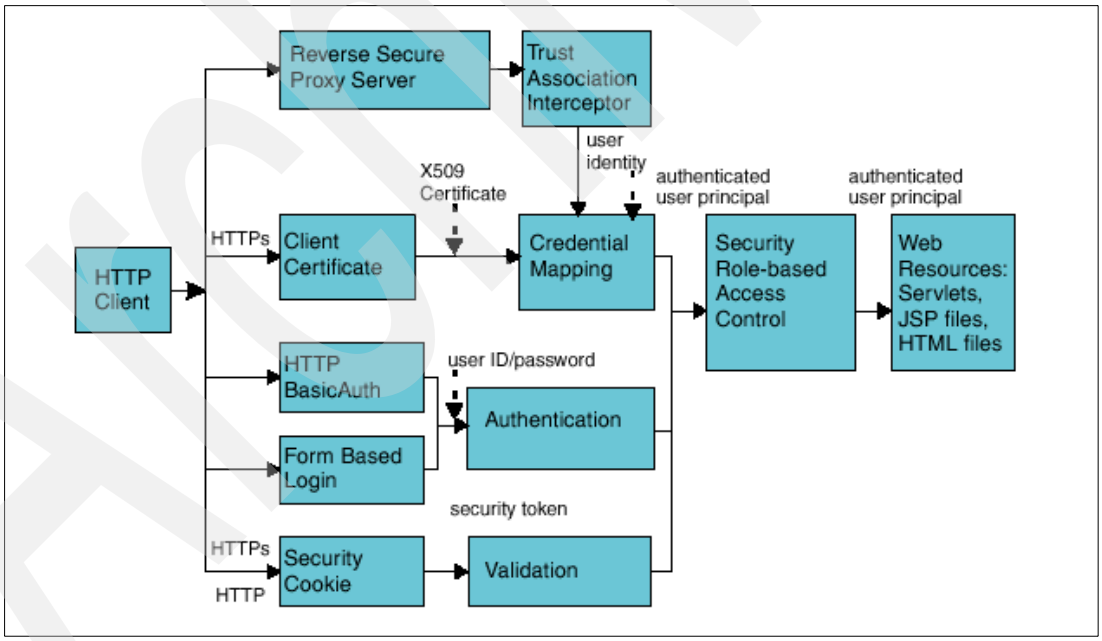


Figure 8-5 Web security

The Web security collaborator enforces role-based access control by using an access manager implementation. An access manager make authorization decision based on security policy derived from the deployment descriptor. An authenticated user principal is allowed to access the requested Servlet or JSP file if it has one of the required security roles. Servlets and JSP files can use the HttpServletRequest methods isUserInRole and getUserPrincipal. As an example, the administrative console uses the isUserInRole method to determine the proper set of administrative functionality to expose to a user principal.

When a servlet or JSP file accesses EJB methods, either the caller identity or a *Run As* identity is propagated to the EJB container, depending on the *Run As* configuration.

EJB security

When security is enabled, the EJB container will enforce access control on the EJB method invocation. The authentication would take place regardless of whether method permission was defined for the specific EJB method.

A Java application client can provide the authentication data in several ways. Using the sas.client.props properties file, a Java client can specify whether to use the user ID and password to authenticate, or to use the SSL client certificate to authenticate. The client certificate is stored in the key file or in the hardware cryptographic card as defined in sas.client.props. The user ID and password may be optionally defined in the sas.client.props file. At runtime, the Java client may perform a programmatic login or a *lazy* authentication.

In lazy authentication, when the Java client is accessing a protected EJB for the first time, the security runtime would try to obtain the required authentication data. Depending on the configuration setting in sas.client.props, the security runtime would either look up the authentication data from sas.client.props or prompt the user. Alternatively, a Java client can use programmatic login. The product supports the JAAS programming model and the JAAS login (LoginContext) is the recommended way of programmatic login. The helper function, *login_helper/request_login*, is deprecated in Version 5. Java clients programmed to the login_helper API still can run in this version.

The EJB security collaborator enforces role-based access control by using an access manager implementation. An access manager makes an authorization decision based on a security policy derived from the deployment descriptor. An authenticated user principal is allowed to access the requested EJB method if it has one of the required security roles. EJB code may use the EJBContext methods isCallerInRole and getCallerPrincipal to check the client's credentials.

J2EE Run As specification is at EJB bean level. When Run As identity is specified, it applies to all bean methods. The method level IBM's Run As extension introduced in Version 4.0 continues to be supported in this version.

8.1.3 WebSphere Application Server administrative roles

WebSphere extended security role based access control to administrative resources includes the JMX system management subsystem, user registry, and JNDI name space. WebSphere administrative subsystem defines four administrative security roles:

- *Monitor* role, which can view configuration information and status but not anything more.
- *Operator* role, which is a monitor that can trigger runtime state changes, such as start an application server or stop an application, but cannot change configuration.
- *Configurator* role, which is a monitor that can modify configuration information but cannot change runtime state.
- *Administrator* role, which is an operator as well as a configurator.

You can perform most administrative work, including installing new applications and application servers, if you have a role as a configurator. There are certain configuration tasks that a configurator does not have sufficient authority to do when global security is enabled:

- Modifying WebSphere Application Server server identity and password
- LTPA password and keys
- Assigning users to administrative security roles

Those sensitive configuration tasks require the administrative role.

WebSphere Application Server administrative security is enforced when global security is enabled. It is recommended that WebSphere Application Server global security be enabled to protect administrative subsystem integrity. For WebSphere Application Server Network Deployment V5.0 for iSeries, the application server security can be selectively disabled if there is no sensitive information to protect.

For information about configuring the administrative security roles in WebSphere Application Server, see 8.2, “Enabling global security” on page 355.

8.1.4 WebSphere Application Server security policies

WebSphere Application Server Java 2 Security implementation was based on the J2EE 1.3 Specification. The Specification granted Web components read and write file access permission to any file in the file system, which may be too broad. WebSphere Application Server default policy gives Web components read and write permission to the subdirectory and the subtree where the Web module was installed. The default Java 2 security policy for all Java virtual machines and WebSphere Application Server server processes are contained in the java.policy file (the default policy for the Java virtual machine) and the server.policy file (the default policy for all product server processes).

To simplify policy management, WebSphere Application Server policy is based on resource type rather than code base (location). There are default policy files for embedded resources, for libraries shared by multiple applications, and for J2EE applications:

- *spi.policy*, which is for embedded resources defined in resources.xml, such as JMS, JavaMail and JDBC drivers
- *library.policy*, which is for shared libraries that are defined by the administrative console
- *app.policy*, which is the default policy for J2EE applications

In general, applications should not require more permissions to run than those recommended by the J2EE Specification, so they can be portable among application servers. Some applications, however, may require more permissions. WebSphere Application Server allows a policy file (was.policy) for each application, which can be packaged with the application and grants extra permissions to that application. Note that granting extra permissions to an application should be handled with great care because of the potential of compromising system integrity.

WebSphere Application Server uses a permission filtering policy file to alert users when an application requires permissions that are on the filter list during application installation, and could cause the offending application installation to fail.

For example, the java.lang.RuntimePermission *exitVM* permission should not be given to an application so that no application code is allowed to terminate the WebSphere Application Server application server. The filtering policy is defined by the filterMask element in the *filter.policy* file.

WebSphere Application Server also performs runtime permission filtering based on the runtime filtering policy to ensure that no application code has been granted any permission that is considered harmful to system integrity. Applying the Java 2 Security model to application servers is new.

WebSphere Application Server runtime uses Java 2 Security to protect sensitive runtime functions. It is always a good idea to enforce Java 2 Security. Applications that are granted with AllPermission have access to sensitive system resources and can potentially cause damage. In cases where an application can be trusted to be safe, WebSphere Application Server allows Java 2 Security to be disabled on a per application server basis.

The global security configuration and Java 2 Security configuration are stored in a set of XML configuration files. Both role-based access control and Java 2 Security permission-based access control are employed to protect the integrity of the configuration data. System integrity is maintained in these ways:

- When Java 2 Security is enforced, application code cannot access the WebSphere Application Server runtime classes that manage the configuration data unless the application code has been granted the required WebSphere Application Server runtime permissions.
- When Java 2 Security is enforced, application code cannot access the WebSphere Application Server configuration XML files unless it has been granted the required file read and write permissions.
- The JMX administrative subsystem provides SOAP over HTTP(S) and RMI/IIOP remote interface to allow application programs to extract and to modify configuration files and data. When global security is enabled, an application program can modify WebSphere Application Server configuration, provided that the application program has presented valid authentication data and that the security identity has the required security roles.
- If a user is allowed to disable Java 2 Security, then that user can modify WebSphere Application Server configuration including the WebSphere Application Server security identity and authentication data. Hence, only users with the administrator security role are allowed to disable Java 2 Security.
- Because WebSphere Application Server security identity is given the administrator role, only users with administrator role are allowed to disable global security, to change server id and password, and to map users and groups to administrative roles.

Other WebSphere Application Server runtime resources are protected by similar mechanisms. It is very important to enable WebSphere Application Server global security and to enforce Java 2 Security.

8.1.5 Backward compatibility

While adding new security functions and moving towards new industry standards, WebSphere Application Server V5.0 for iSeries maintains backward compatibility with the 4.0.x and 3.5.x releases. Applications created in the Version 4.x development environment can be deployed in Version 5. When Java 2 Security is enforced in Version 5, special consideration needs to be given to Version 4.0.x applications because Version 4.0 applications may not be Java 2 Security ready.

WebSphere Application Server Version 4 supports Java 2 Security, but only enforces permission checks against *exitVM*, and creating and configuring the Security Manager. Other permission checks are disabled by default. Hence, many applications developed for prior releases of WebSphere Application Server may not be ready for Java 2 Security.

8.2 Enabling global security

In this section you will learn how to enable global security within WebSphere. You will see how to configure access to the Administrative console by defining a number of roles and mapping those roles to the existing users. Then, in order to test your configuration, you will attempt to perform various functions using the various users.

You will see if the security configurations correctly limit your ability to perform these functions:

1. Verify that your WAS instance has been started.
2. Start the Admin Console and view local configuration information:
   a. In a browser, specify the URL `http://<hostName>:port/admin`

      Here, <hostName> is the name of your iSeries where you run a WAS instance for which you want to enable the global security. port is the port number on which the admin console is listening for the requests.

   b. In the Login window, specify your user ID for **User ID** and click **OK**. At this stage user ID can be any string.
3. Click the plus sign (+) to the left of **Security** to expand the tree. Now expand the **User Registries** and click **Local OS**.
4. In the right pane we need to specify a userID and password (need to be for a privileged user), so that WebSphere can authenticate against the local registry. Enter your iSeries user ID as **Server User Id** and your password on this iSeries for **Server User Password** (see Figure 8-6).

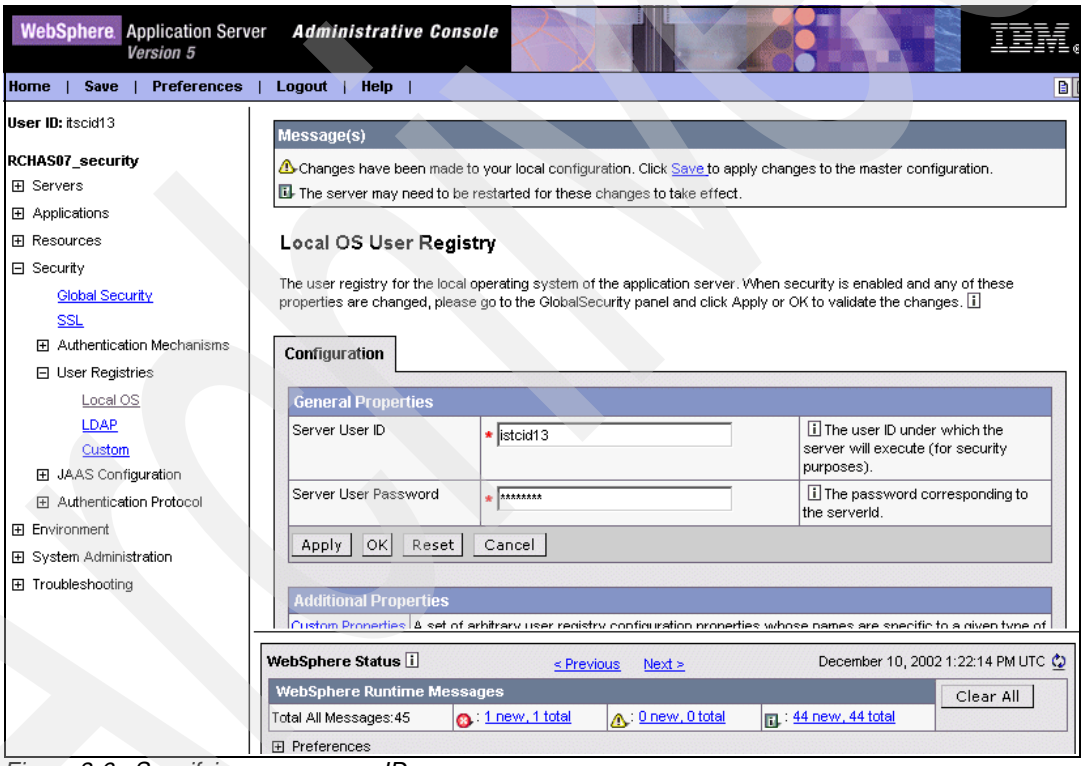


Figure 8-6 Specifying secure user ID

5. Click **OK**.
6. A new frame appears. In the right pane, look for the checkbox labeled **Enabled** and click it. This will check the **Enforce Java 2 Security** box automatically. Ensure that **Active User Registry** is set to **LocalOS** (see Figure 8-7).

Important: In our example we use SWAM as the **Active Authentication Mechanism**. Only SWAM can be used with the Base Edition of WAS. Lightweight Third Party Authentication (LTPA) must be used to enable global security for the Network Deployment Edition.

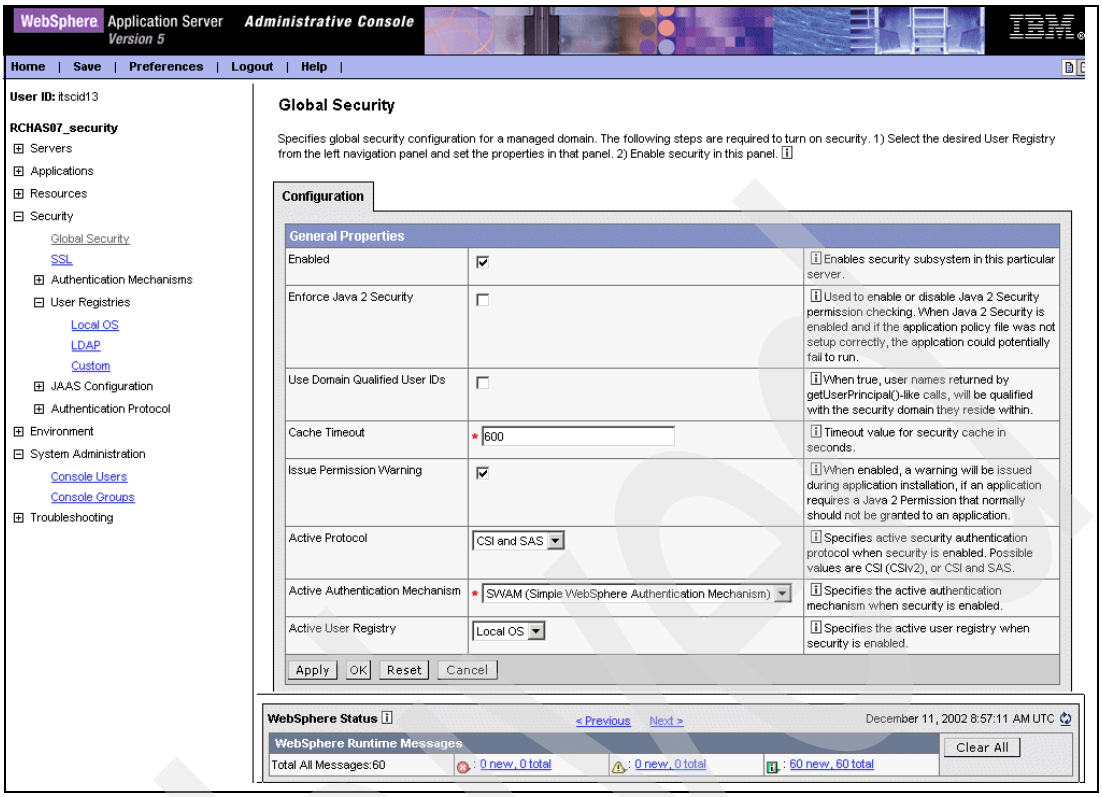


Figure 8-7 Enabling global security

7. Click **OK**.

   Be sure to check the message panel at the top of the page. Any errors should be reviewed and corrected before continuing (see Figure 8-8).

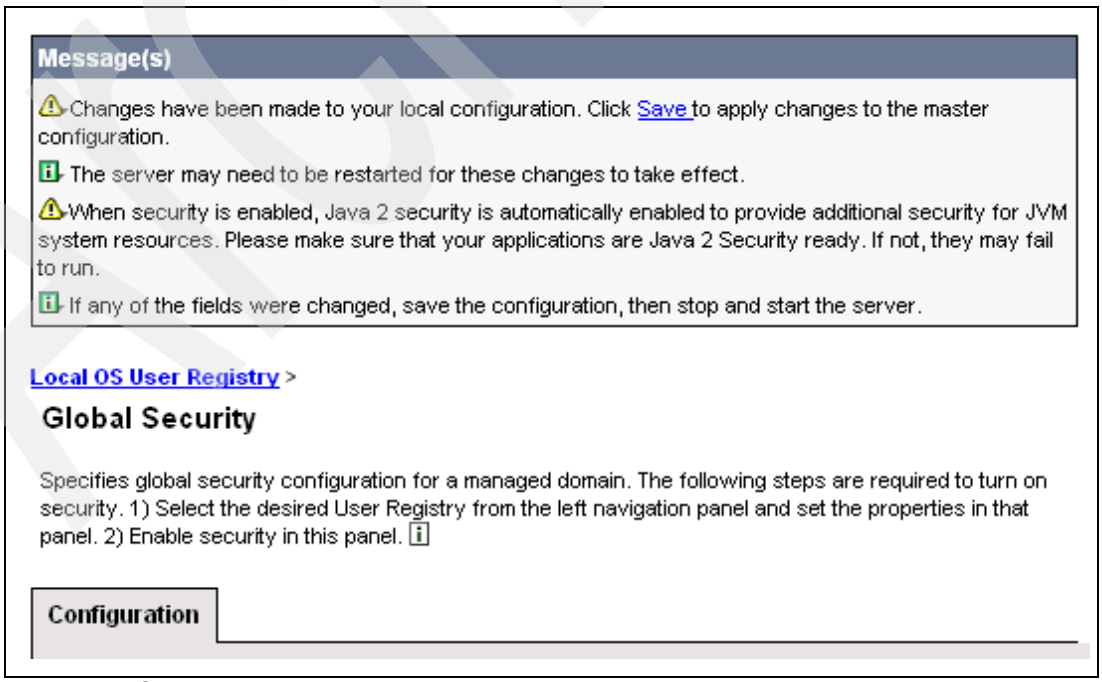


Figure 8-8 Checking the messages

8. When no errors are shown in the message panel, click **Save** in the menu bar, then click the **Save** button to save the master configuration.
9. From the Admin Console, expand **System Administration** in the topology tree and select **Console Users**.
10. Click **Add**.
11. Now you can map a user ID on your iSeries (because we use Local OS user registry) to each of four predefined administrative security roles (see the description of each role in “WebSphere Application Server administrative roles” on page 353):
    - Administrator (the user ID that you specified while enabling global security in step 4 on page 356 has the Administrator security role’s access rights)
    - Configurator
    - Operator
    - Monitor
12. Enter a valid iSeries user ID as **User** and click one of the **Role(s)**.

Tip: The best practice for mapping a user to a role is to create an OS/400 group profiles for each of four administrative roles. Then, managing access to the administrative functions of WAS is done via membership in these group profiles.

There are two advantages to be gained by using this technique:

- It’s easier to add a new user profile to an OS/400 group profile
- You don’t have to restart an application server after adding a new user profile to one of the 4 administrative roles

13. Click **OK**.
14. Map as many user IDs to the different security roles as needed.

Figure 8-9 Mapping predefined roles to the user IDs

15. After finishing this mapping, you should see a screen similar to Figure 8-10.

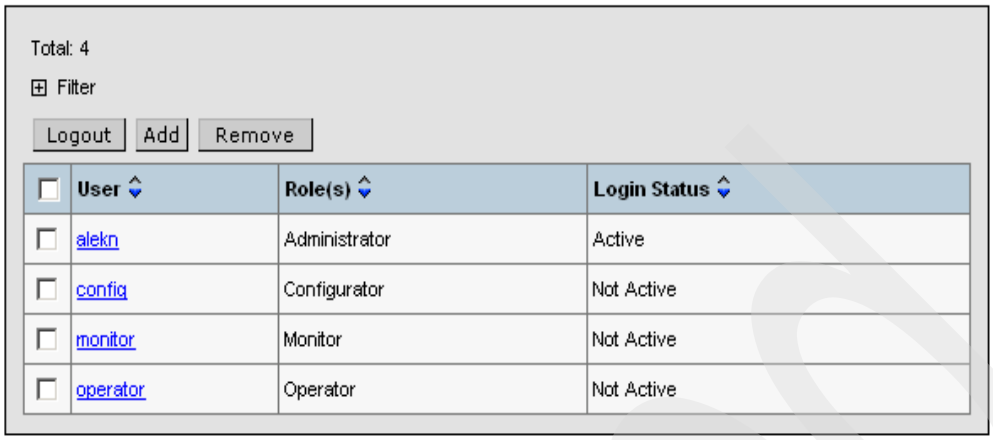


Figure 8-10 Created console users

16. Save your changes.
17. Now click **Logout** in the menu bar.
18. In order for the Global Security to take effect, stop and start your WebSphere Application Server instance:
    a. From the Qshell command line, execute the following two commands:

```
cd /qibm/proddata/webas5/base/bin
stopServer -instance instanceName serverName
```

Here, `instanceName` is the name of your WAS instance for which you've enabled global security, and `serverName` is the name of the application server in the `instanceName` instance.

> **Note:** Server name is case sensitive.

    b. Wait for the completion of the last command and start your instance again:

```
startServer -instance instanceName serverName
```

> **Important:** Next time you need to use the `stopServer` script, after restarting an application server with enabled security, you will have to specify a user ID and password (these parameters have been set in step 4 on page 356). For example:
>
> `stopServer -instance instanceName serverName -username myId -password myPassword`

19. Wait for the confirmation message that the server has been started.
20. Point your browser to the following URL:

```
http://<hostName>:port/admin
```

21. You should see a login prompt with two boxes: User ID and Password (see Figure 8-11).

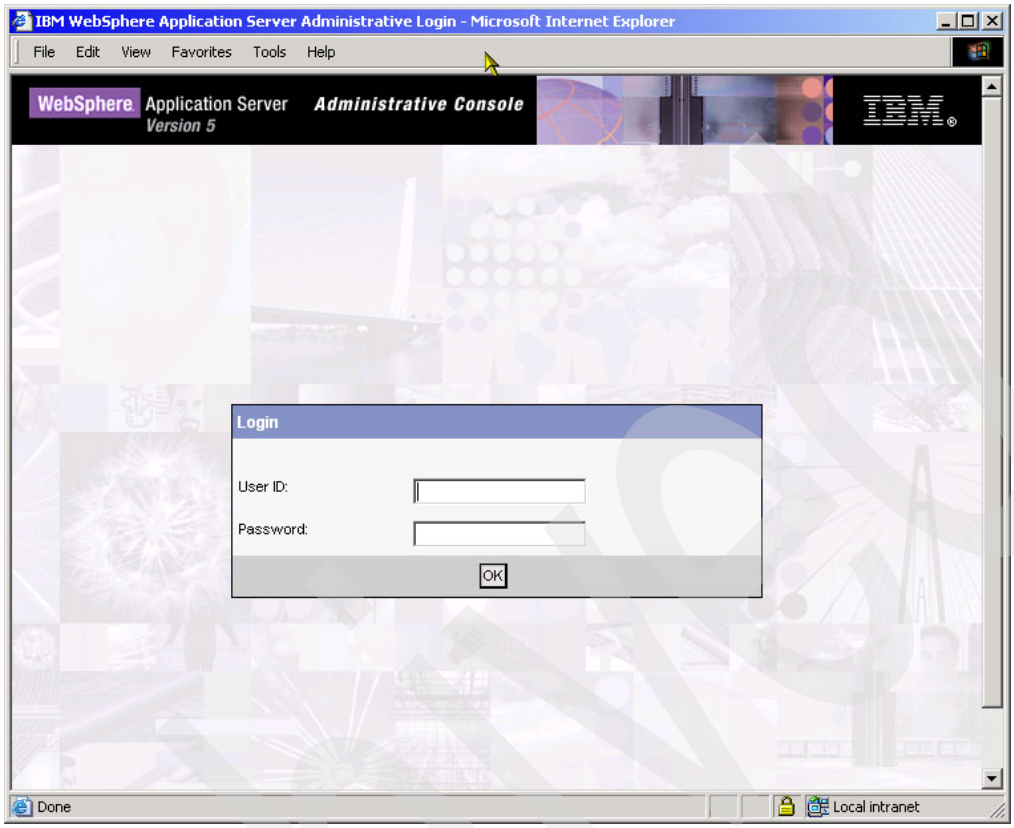


Figure 8-11 Login prompt

22. Now you can test access through four administrative security roles that you've defined in step 11 on page 358:

    - Administrator
    - Configurator
    - Operator
    - Monitor

23. Notice that each user's console will have different functions available. Go ahead and experiment with each one, starting, stopping, and exporting various applications:

    i. Start a new Admin Console and authenticate with the `Monitor` user ID. Notice that when you look at **Enterprise Applications** under **Applications**, you have very few actions enabled. You cannot even start or stop an application. The only two functions listed are **Export** and **Export DDL** (see Figure 8-12).

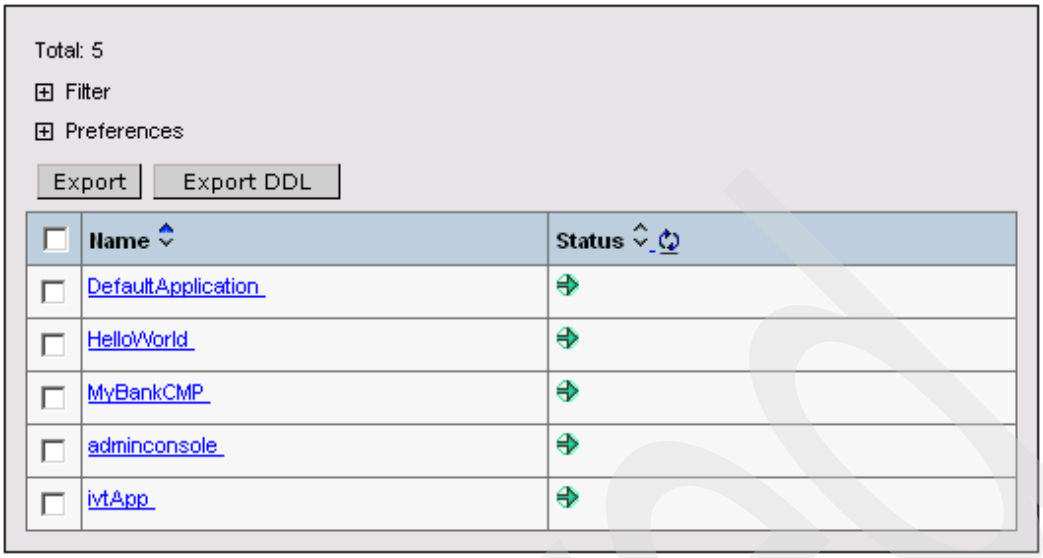


Figure 8-12 Options available for user ID "monitor"

ii. Start another Admin Console session and authenticate with the `Administrator` role. Notice that when you look at **Enterprise Applications** under **Applications** that you now have much more functionality available (see Figure 8-13).

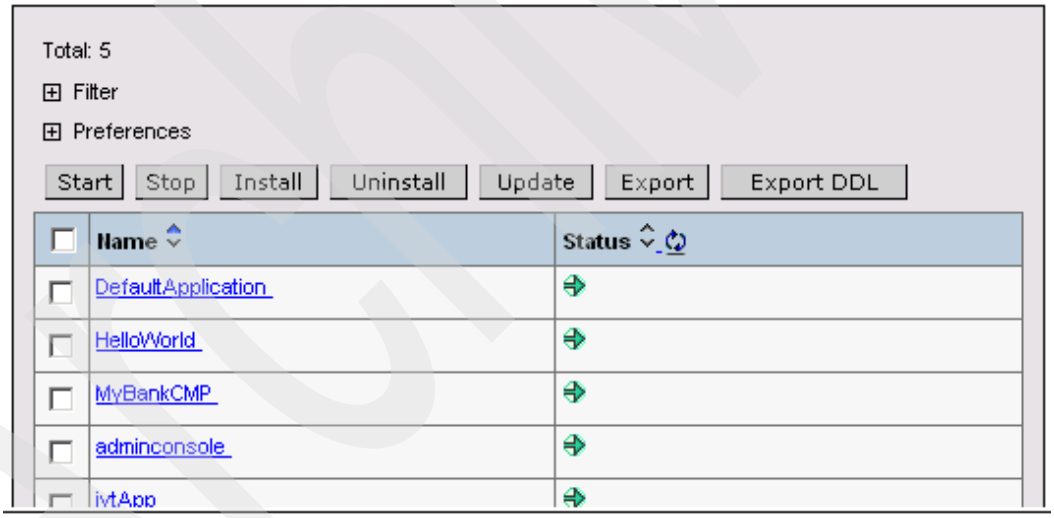


Figure 8-13 Options available for user ID "Administrator"

8.3 Enable SSL

In this section we discuss how to:

- Create your own SSL certificate
- Install the certificate with the IBM HTTP Server (powered by Apache)
- Change your WebSphere Application Server V5.0 for iSeries configuration to accept SSL requests

8.3.1 Creating your own SSL certificate

Creating your own SSL certificate is a two-step process. In the following section we cover these tasks.

Creating a local Certificate Authority

A Certificate Authority (CA) is a central administrative entity that can issue digital certificates to users and servers. The Certificate Authority *signs* certificates with its private key to validate their authenticity. A CA can be a publicly available entity, such as VeriSign, or it can be a privately created entity, such as a private intranet or local CA. Digital Certificate Manager (DCM) allows you to use both types of CAs to request and install certificates.

There can only be one CA per iSeries server. In case one has already been created on your system, you can continue with 8.3.3, “Creating a server certificate with a local CA” on page 370.

Follow these steps to set up the Certificate Authority:

1. Logon to the iSeries Tasks page by entering the URL:

   http://<Your iSeries Server>:2001

2. Sign on with your user ID and password:

   Note that the user profile needs to have *ALLOBJ and *SECADM special authorities.

3. Click the **Digital Certificate Manager** link. The window shown in Figure 8-14 appears.

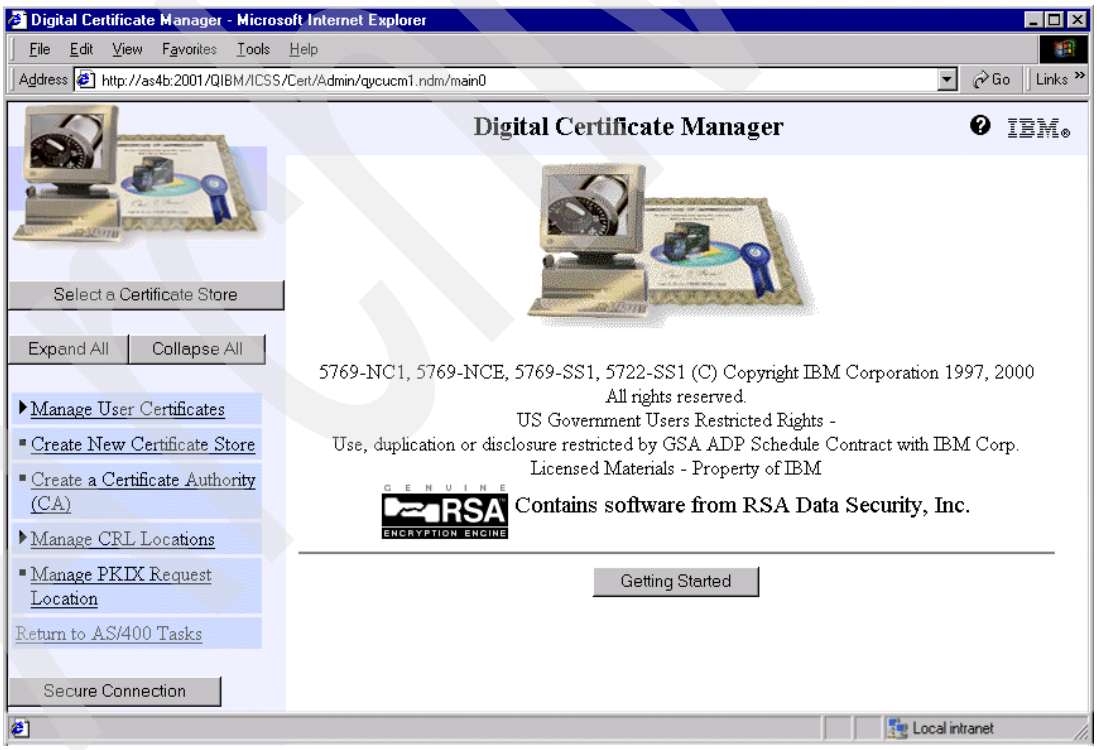


Figure 8-14 Digital Certificate Manager

4. Click **Create a Certificate Authority (CA)**. The window shown in Figure 8-15 appears.

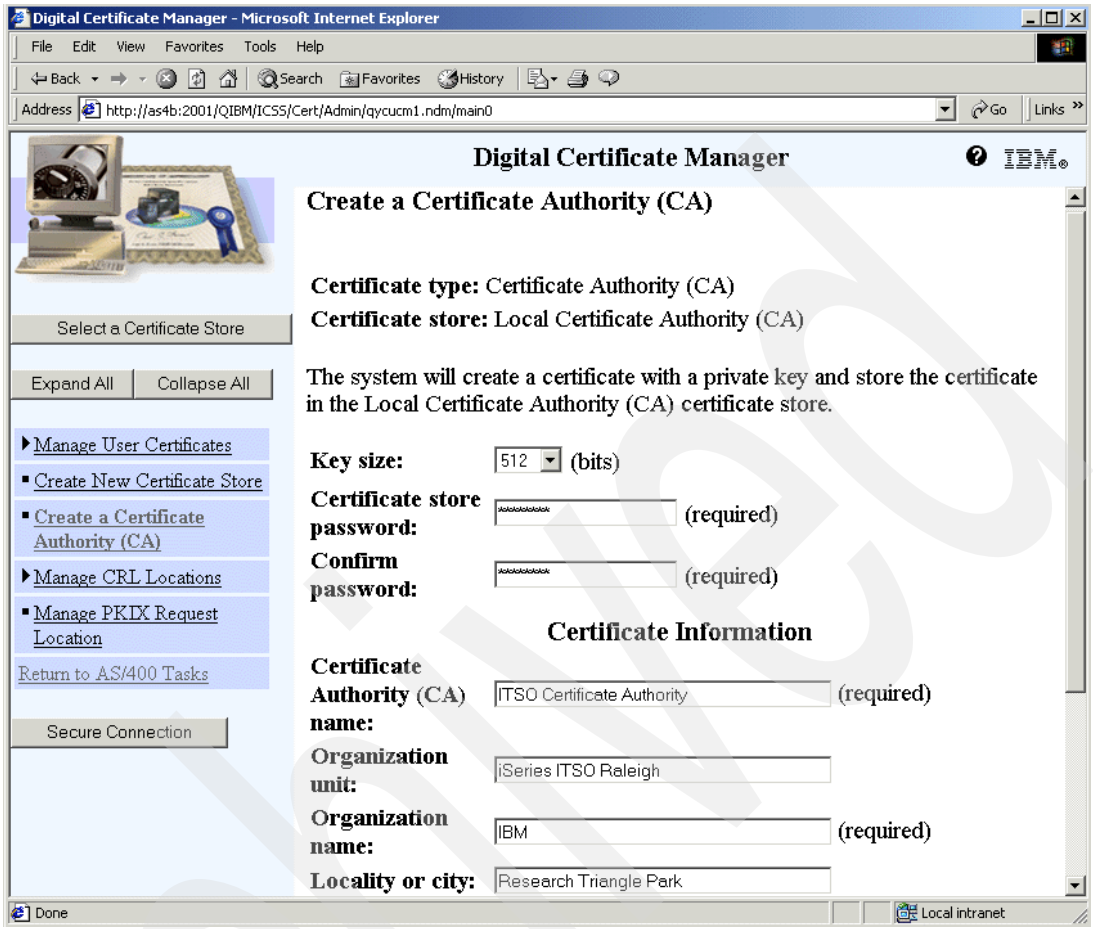


Figure 8-15 Creating a Certificate Authority

If you have a 4758 Cryptographic Coprocessor installed and configured, you are initially presented with a selection window to specify where to store the Certificate Authority's private key.

5. Complete the Create a Certificate Authority (CA) form and click **Continue**.

   Digital Certificate Manager processes the form and creates the following directories and files:

   The **/QIBM/UserData/ICSS/Cert/CertAuth** directory contains the following objects:

   – **CA.TXT**: Contains the CA certificate in Base64 encoded form
   – **DEFAULT.KDB**: Contains the private key and the CA certificate
   – **DEFAULT.POL**: Is the CA policy file
   – **DEFAULT.RDB**: Is the CA's request database

   The **/QIBM/UserData/ICSS/Cert/Download/CertAuth** directory contains the **CA.CACRT** binary form of CA certificate.

Next you need to install the CA certificate on your browser. The installation screen appears, as shown in Figure 8-16.

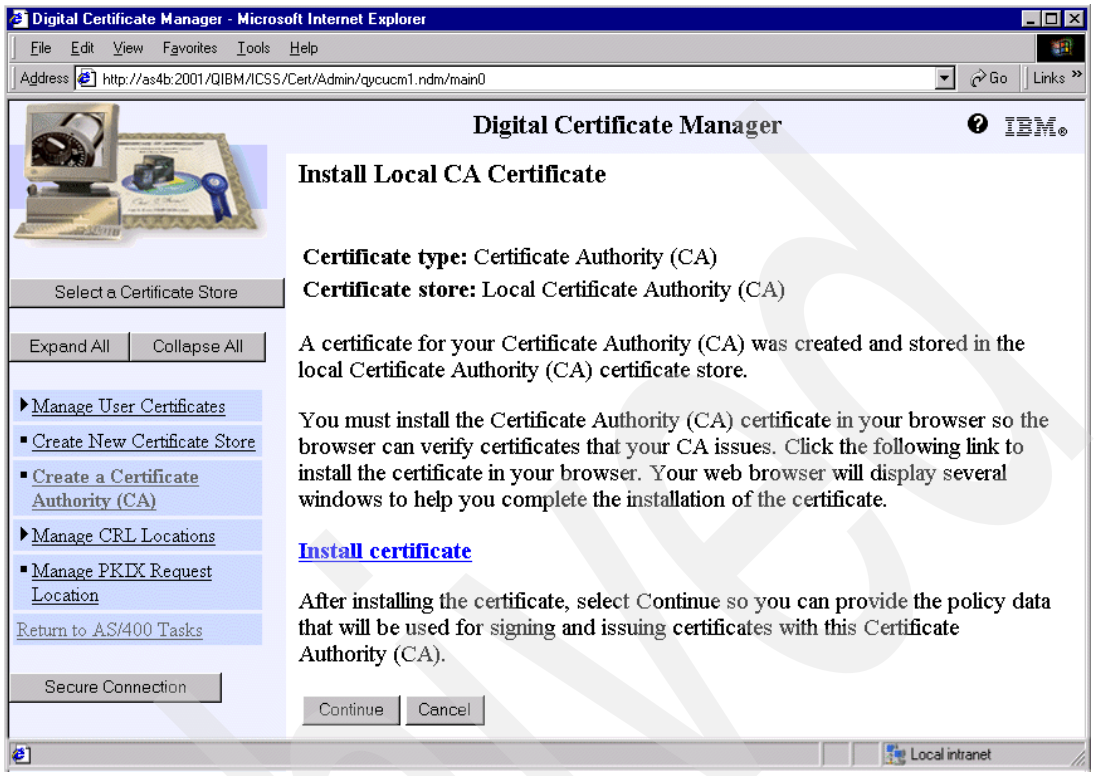


Figure 8-16 Install local CA Certificate

6. Click the **Install certificate** link. Your browser will prompt you with security warnings. Ignore these and click **Open** to begin the installation.
7. Follow the instructions issued by your browser to install the certificate on your PC.
8. Click **Continue**. The Certificate Authority (CA) Policy page appears, as shown in Figure 8-17.

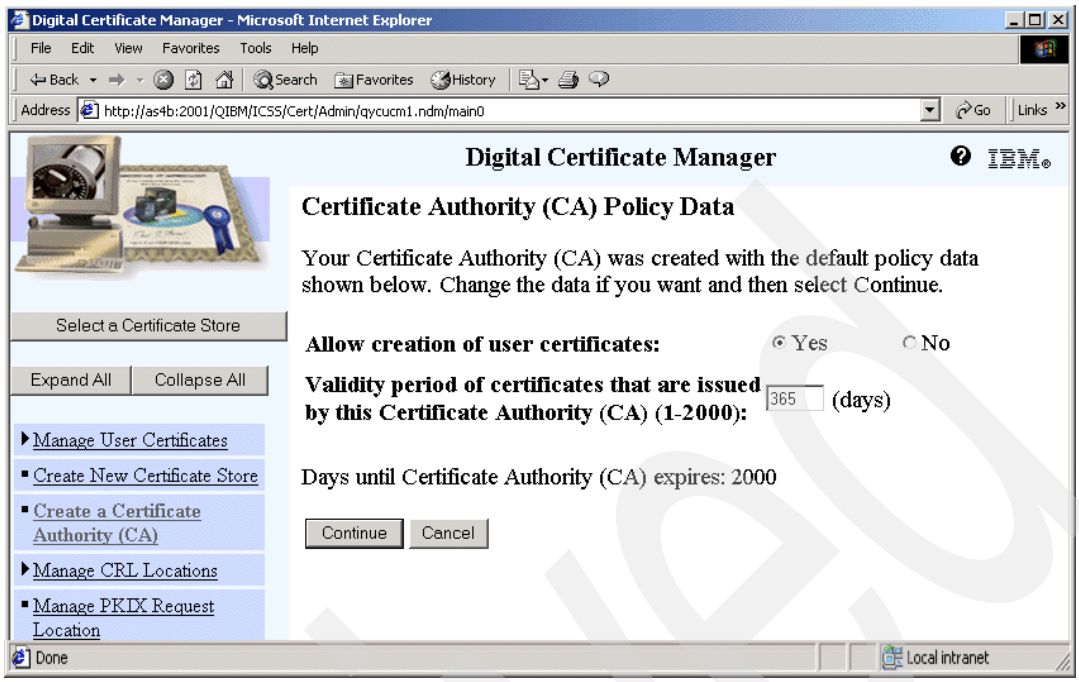


Figure 8-17 Entering policy data

9. Select **Yes** for Allow creation of user certificates, and click **Continue**.

   You have now created a local CA on your system and can use it to issue certificates for server and client applications, object signing, and users.

   Note: When you create a Certificate Authority (CA) with Digital Certificate Manager, you can specify the policy data for the CA. The policy data for a (CA) describes the signing privileges that it has. The policy data determines whether the CA can issue and sign user certificates and how long certificates that the CA issues are valid.

   A confirmation message shown in Figure 8-18 appears.

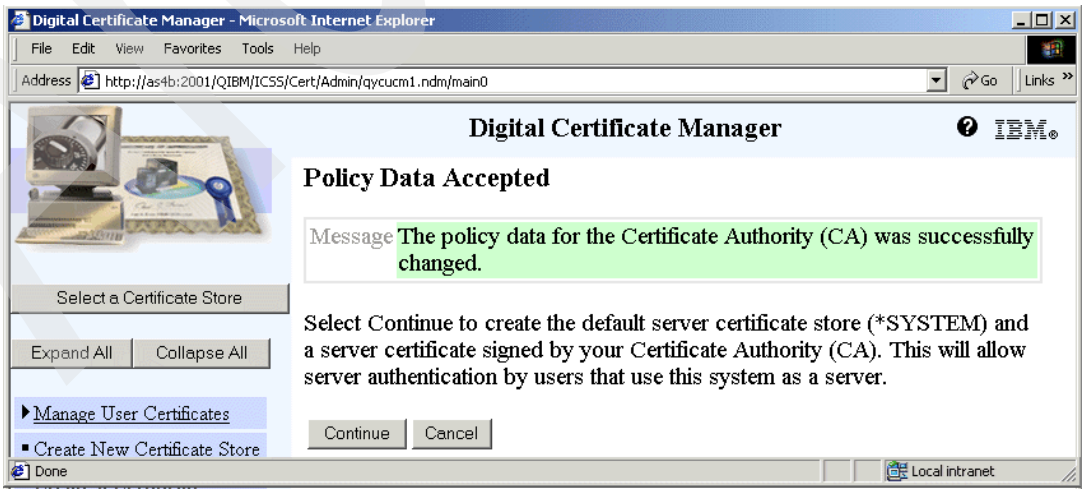


Figure 8-18 Policy data confirmation

10. Click **Continue**.

Create a Server Certificate

The next step is to create a server certificate to be used by the secure HTTP server (and other applications that run over SSL) during the SSL handshake.

If you have a 4758 Cryptographic Coprocessor installed and configured on this system, another window appears asking for the location of the server certificate's private key.

11. Complete the Create a Server or Client Certificate form as shown in Figure 8-19.

Figure 8-19 Create a Server or Client Certificate

12. Click **Continue**.

DCM creates the system certificate in the *SYSTEM certificate store. The *SYSTEM certificate store consists of the following files in **`/QIBM/UserData/ICSS/Cert/Server`**:

DEFAULT.KDB: Contains server and client certificates with their private keys

DEFAULT.RDB: Is the certificate request database

DCM displays the next window to select applications that use this server certificate. At the top of the window a confirmation message appears stating that the server certificate has been created.

13. Click **Continue**. A confirmation button appears, as shown in Figure 8-20.

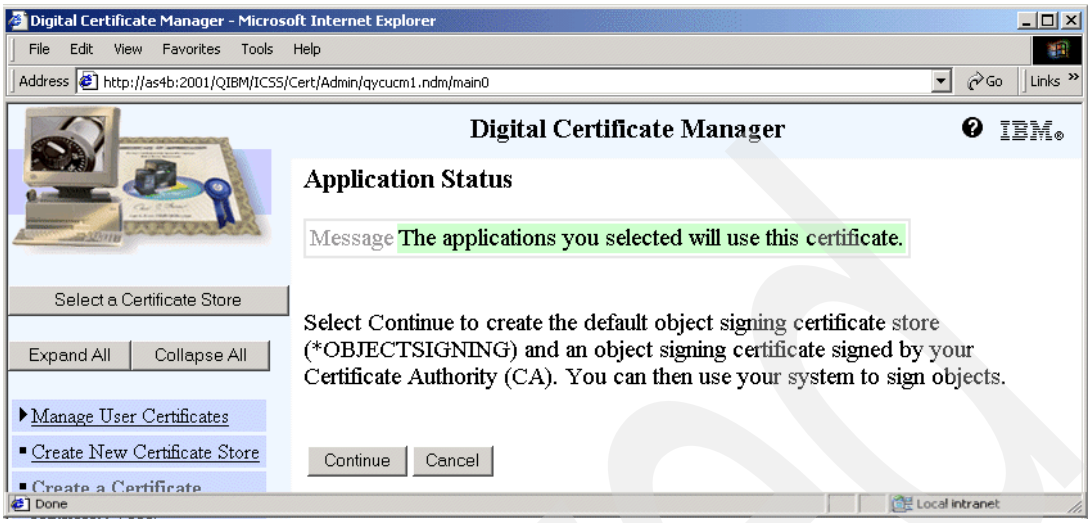


Figure 8-20 Application status

Object Signing store creation

14. Click **Continue**. The window shown in Figure 8-21 appears.

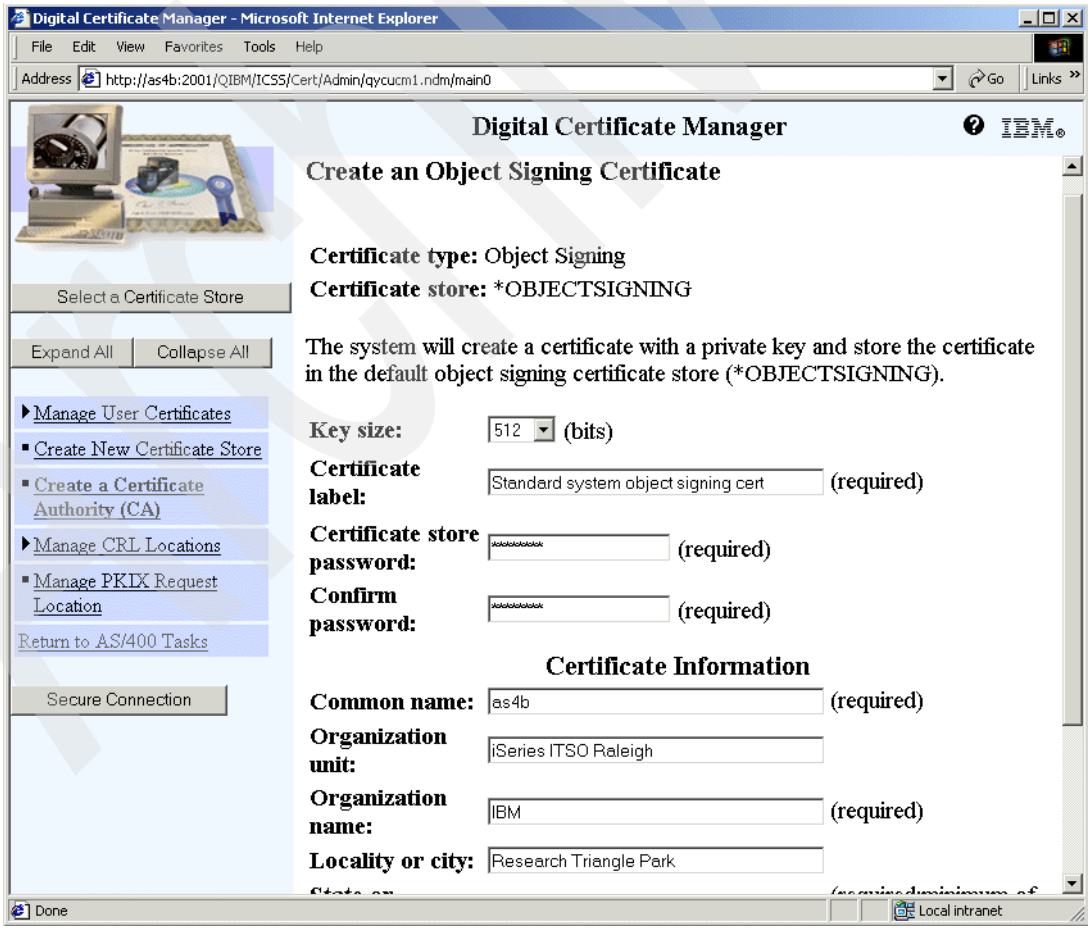


Figure 8-21 Creating an Object Signing Certificate

15. Enter the certificate's RSA key length, object signing certificate store password, and a label for the new object signing certificate. Then, enter the certificate's subject information as you did previously for the server certificate.
16. Click **Continue**. The Select Applications window appears, as shown in Figure 8-22.

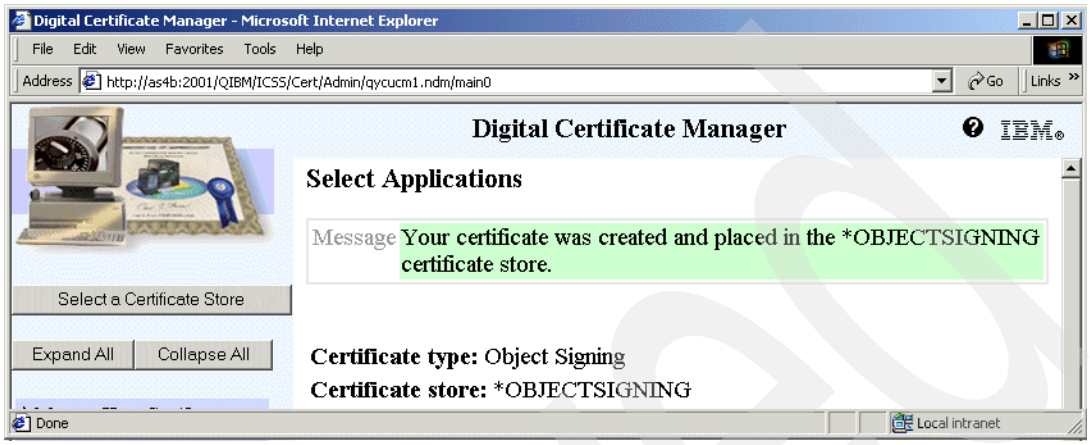


Figure 8-22 Select Applications

The window contains a confirmation message about the creation of the object signing certificate. If you used object signing previously on this system, a list of available object signing applications appears. You can then select the applications for which you want to use the new certificate. In a new system environment, no applications are shown.

17. Click **Continue**.
18. You now must select all the applications that will trust the newly created Certificate Authority. The list contains all registered client applications and the server applications that support client authentication. At this point your HTTP server will not be listed, so simply click **Continue**.
19. A final confirmation message appears, as shown in Figure 8-23. Click **OK** to complete the setup.

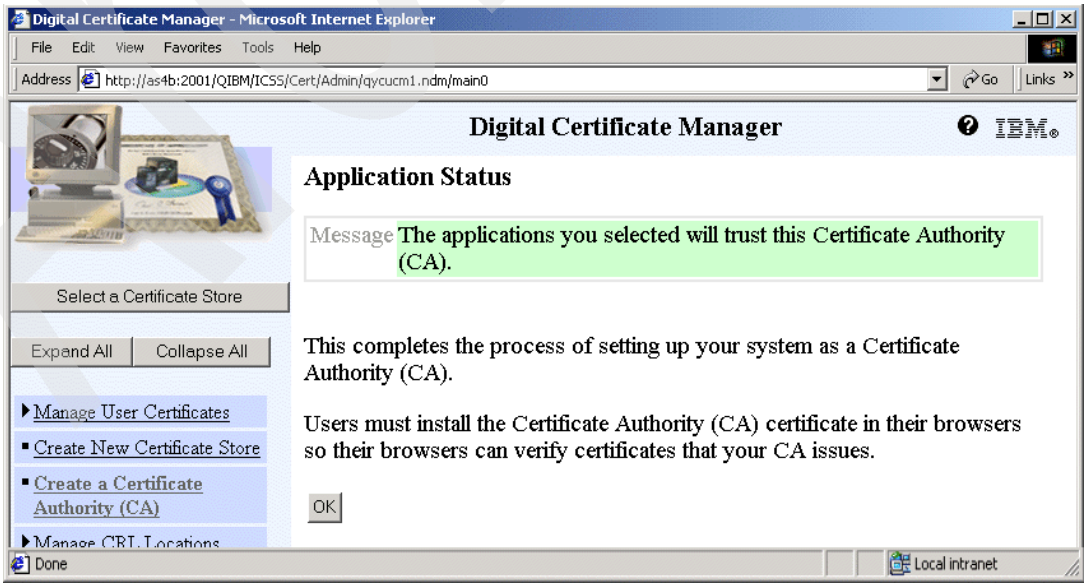


Figure 8-23 Application status

8.3.2 Setting a default certificate label

The first time the local CA and *SYSTEM certificate store are created, there is no default certificate assigned to the certificate store. As long as all SSL applications, whether system provided or user written, specify explicitly a certificate when establishing an SSL session, the default certificate setting is not important. But there are also applications, mostly user written ones, that may not specify a certificate, and then the default certificate is chosen. For this reason it is recommended to set a default certificate label for the *SYSTEM certificate store.

To do this, follow these steps:

1. Click **Select a Certificate Store** on the DCM navigation pane.
2. Select the ***SYSTEM** certificate store and click **Continue**.
3. Enter the certificate store password (see 4 on page 363) and click **Continue**.
4. Click **Fast Path**.
5. From the Fast Path options, click **Work with server and client certificates**. The page appears, as shown in Figure 8-24.

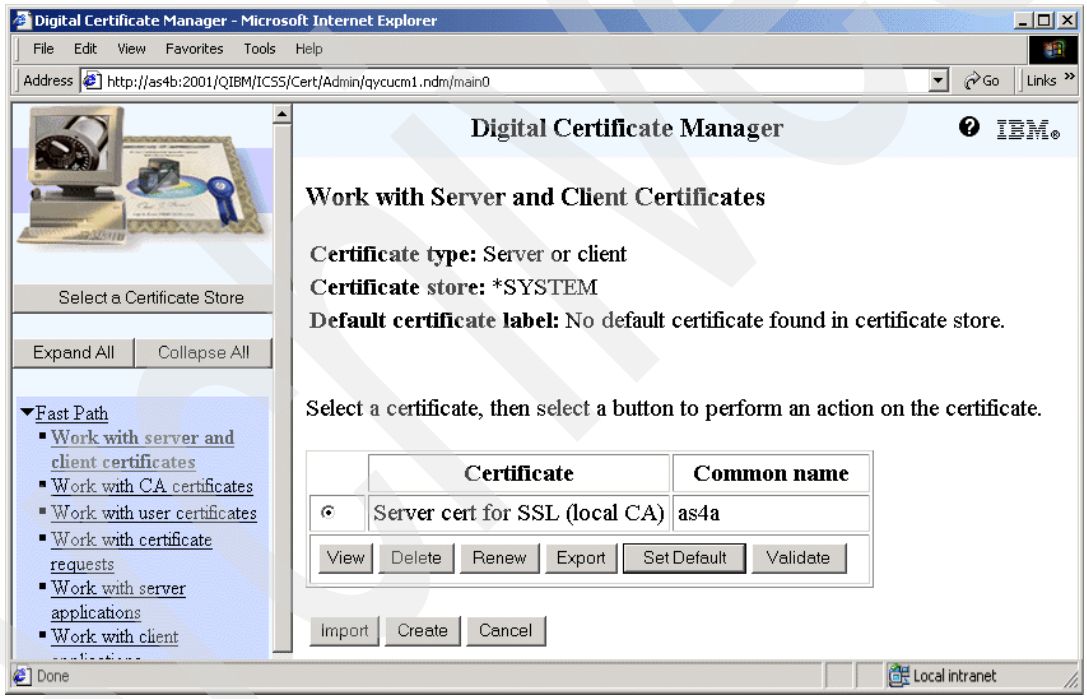


Figure 8-24 Work with server and client certificates

6. To assign a default certificate label to the ***SYSTEM** certificate store, choose the server certificate that you want to use as the default certificate and click **Set Default** as shown in Figure 8-24.

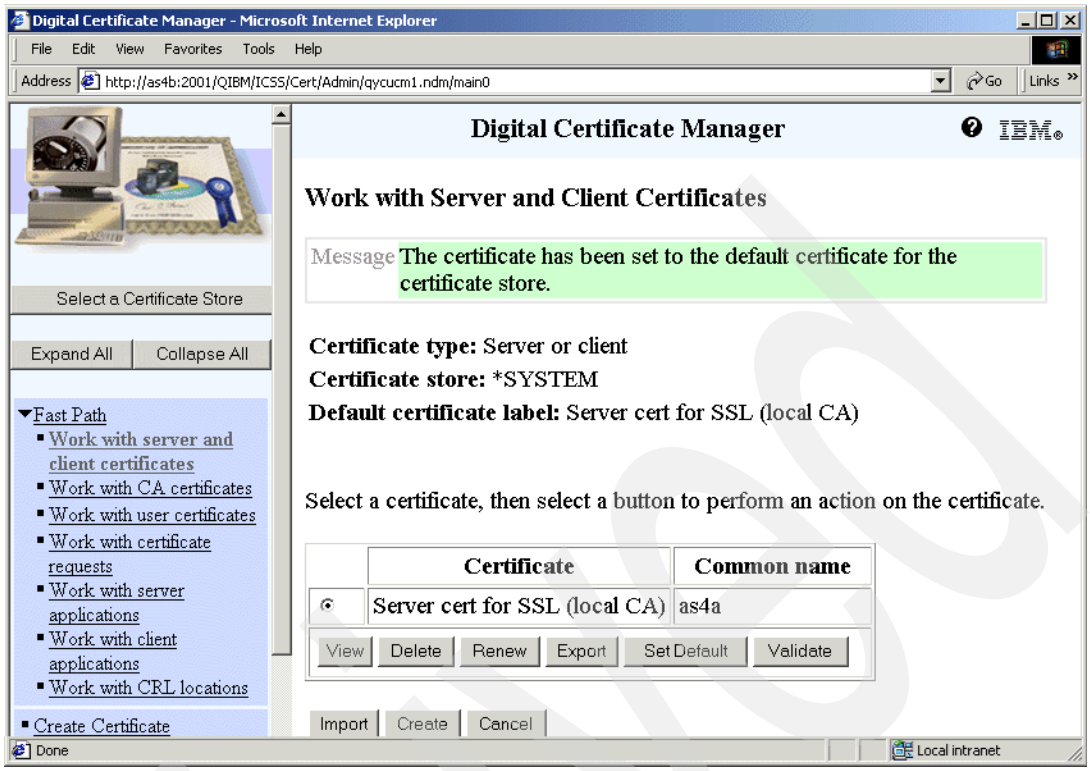


Figure 8-25 Default certificate label

7. Click **Cancel** to complete the setup.

8.3.3 Creating a server certificate with a local CA

A server certificate is a digital document that validates the identity of the system or server that owns the certificate. Server certificates are issued by a Certificate Authority and contain identifying information about the server, such as the server's distinguished name. The certificate also contains the server certificate's public key. A server must have a digital certificate to use Secure Sockets Layer (SSL) for secure communications. Actually the SSL protocol itself also allows SSL sessions without certificate use. However, due to security reasons, this support is not provided in OS/400.

Most iSeries applications, such as the HTTP server, Telnet server, and so forth, support SSL and can examine a server's certificate to verify the identity of the server when the client program requests a secure connection.

We will now create a server certificate to use with our HTTP server instance.

1. Click the **Select a Certificate Store** button on the navigation pane. The window appears, as shown in Figure 8-26.

Figure 8-26 Select a Certificate Store

2. Select the ***SYSTEM** certificate store and click **Continue**. The window shown in Figure 8-27 appears.

   The system certificate store is the standard store for managing server and client certificates used for secure applications.

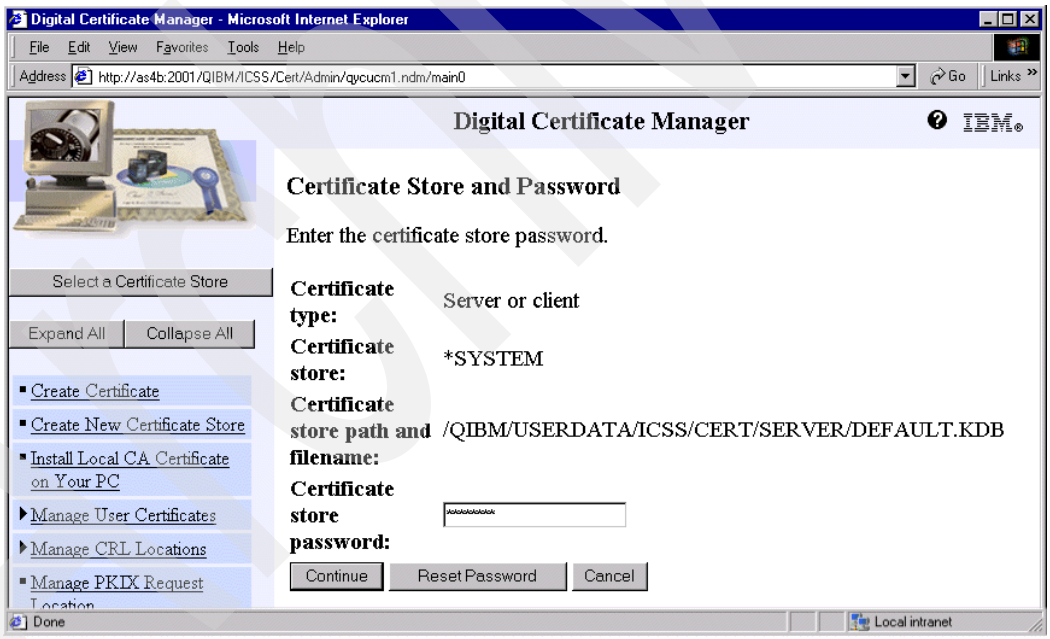


Figure 8-27 Certificate Store and Password

3. Enter the certificate store password and click **Continue**.

Tip: If you have forgotten your certificate store password, you can reset it by clicking **Reset Password**.

The navigation pane refreshes and all of the general options and options available in the system certificate store appear. There are two methods for navigating through the redesigned DCM interface. You can use the individual links and sub-menus on the navigation pane, or you can use the Fast Path option.

4. Click **Fast Path** on the navigation pane.
5. Click **Work with server and client certificates** under the Fast Path navigation bar menu. The window appears, as shown in Figure 8-28.

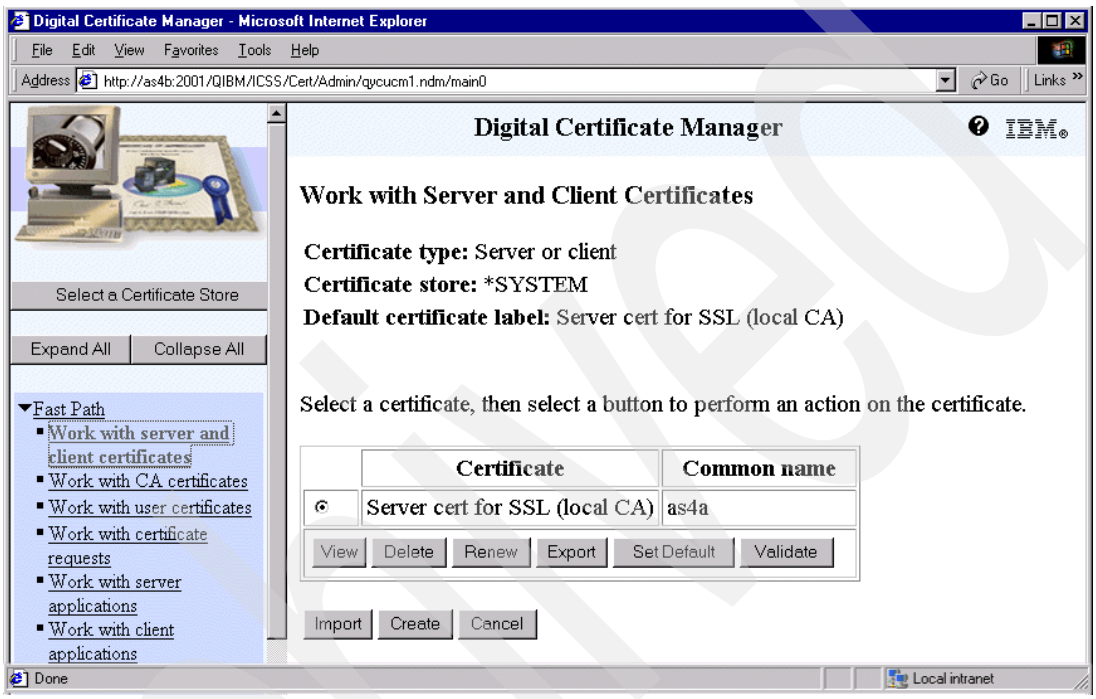


Figure 8-28 Work with certificates

Using the Fast Path method to perform DCM management tasks allows you to perform all tasks related to a category in one place. For example, the Work with Server and Client Certificates window shows all of the options you need to perform certificate management tasks. Note that the certificates for client and server applications are the same. However, the distinction is made when defining the secure application.

6. Click **Create** to create a new certificate. The window shown in Figure 8-29 appears.

Figure 8-29 Select a Certificate Authority (CA)

The number of certificate authorities displayed depends on the selected configuration method:

- **Local Certificate Authority (CA)**: This option is only available when a local certificate authority has been created on the system. When you create the certificate request, the signing request is automatically signed by the local CA and the signed certificate is imported into the system certificate store. The private key that is created when requesting a certificate is stored in the system certificate store and does not leave the store.
- **VeriSign or other Internet Certificate Authority (CA)**: Use this option to create a certificate signing request (CSR). The private key is generated and then stored in the system certificate store. The CSR is sent to the Internet CA, which signs the certificate and then sends the signed certificate back. The signed certificate must then be imported into the system certificate store. This option is always available.

If you have a 4758 Cryptographic Coprocessor installed and configured on the system, another window appears prompting you for the location where the private key should be stored. This option is only available when you request a certificate from a local CA or an Internet CA.

7. Select **Local Certificate Authority (CA)** and click **Continue**. The window shown in Figure 8-30 appears.

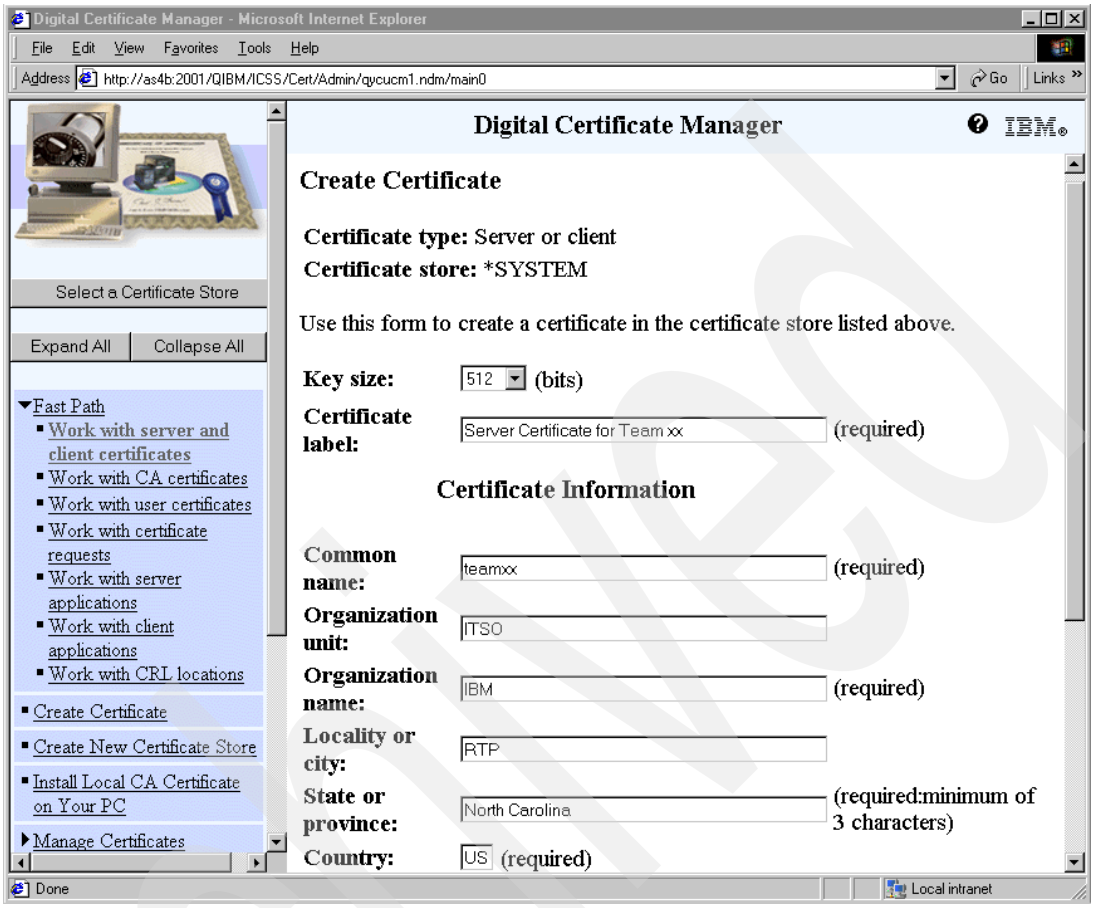


Figure 8-30 Create certificate form

8. Complete the form with information relevant to your company and click **Continue**. The application selection screen appears, as shown in Figure 8-31.

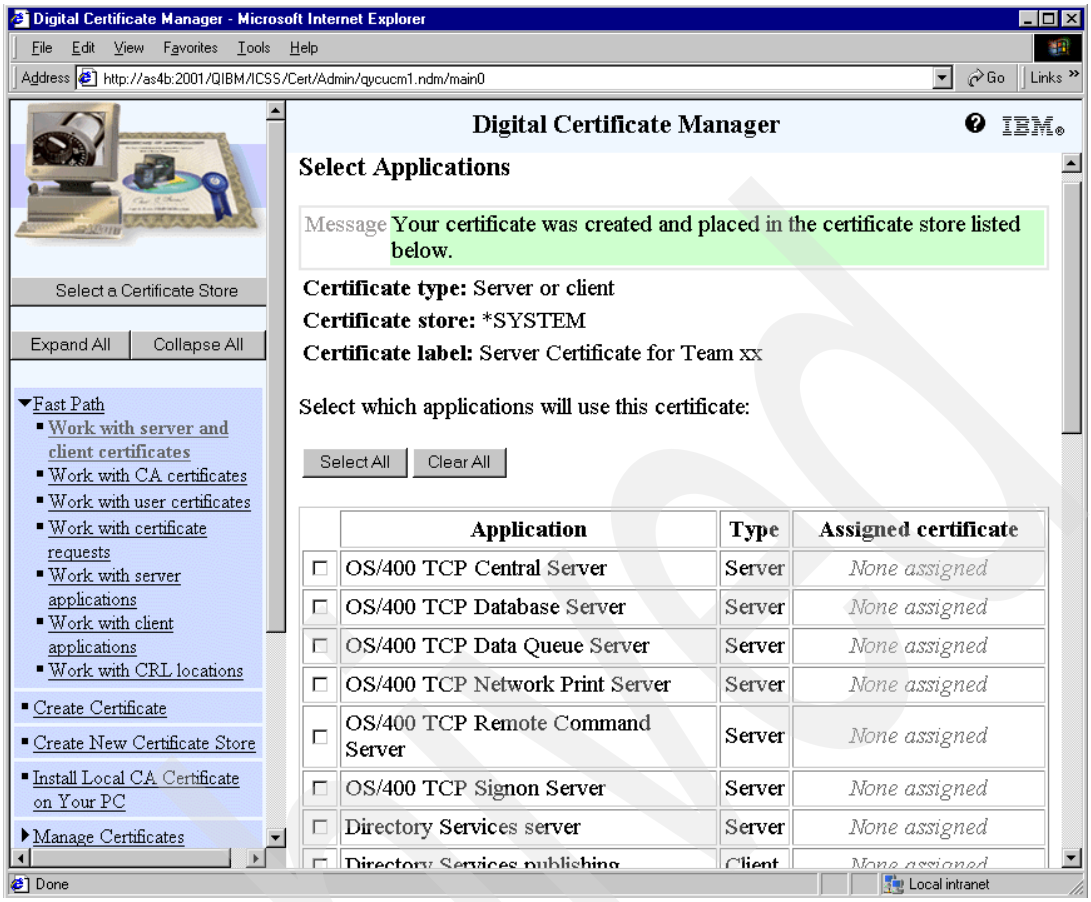


Figure 8-31 Select applications for the certificate.

9. Do not select any applications. Just click **Cancel**.

8.3.4 Enabling SSL with your WebSphere Application server instance

To enable SSL access to your WebSphere Application Server V5.0 server instance, you need to configure the HTTP server to use SSL and add a virtual host entry to your WebSphere Application Server V5.0. In the following sections we cover these steps.

Note: This section only provides instructions for enabling SSL between the Web browser and the HTTP server. SSL can also be enabled between the HTTP server and WAS. Administrators should determine if this extra protection is desirable for their installations before enabling SSL between the HTTP servers and the application servers. For more information, see:

`http://publib.boulder.ibm.com/iseries/v1r1m0/websphere/ic2924/info/rzaiz/50/sec/seccssl.htm`

Enabling SSL in your HTTP server instance

We will now enable SSL for your HTTP server instance. This requires a number of steps as we will need to allocate a new virtual host to handle the SSL connections, create a new server certificate application and assign a certificate to that application.

Follow these step to enable SSL in your application:

1. Start by launching your Web browser and point it to the HTTP Server administration URL:

   `http://[your iSeries System]:2001`

2. Sign on using your iSeries User ID and password and click **IBM HTTP Server for iSeries**.
3. Click on the **Manage** tab and select your HTTP server instance.
4. On the task bar, click **General Server Configuration**. The configuration settings appears, as shown in Figure 8-32.

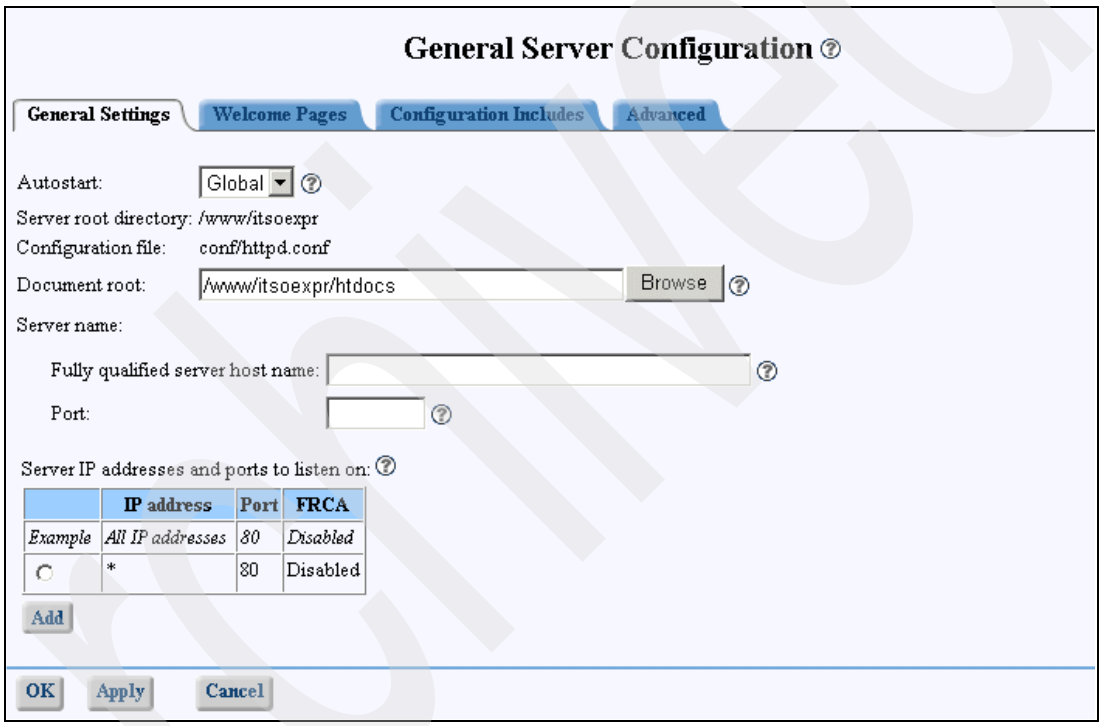


Figure 8-32 General server configuration

5. Click on the **Add** button to add a new port to listen on.
6. Add the SSL port (normally 443) and click **Apply**.

7. In the task bar, click **Virtual Hosts**, then the **IP Based** tab. Add a new virtual host for your SSL port (443) as shown in Figure 8-33.

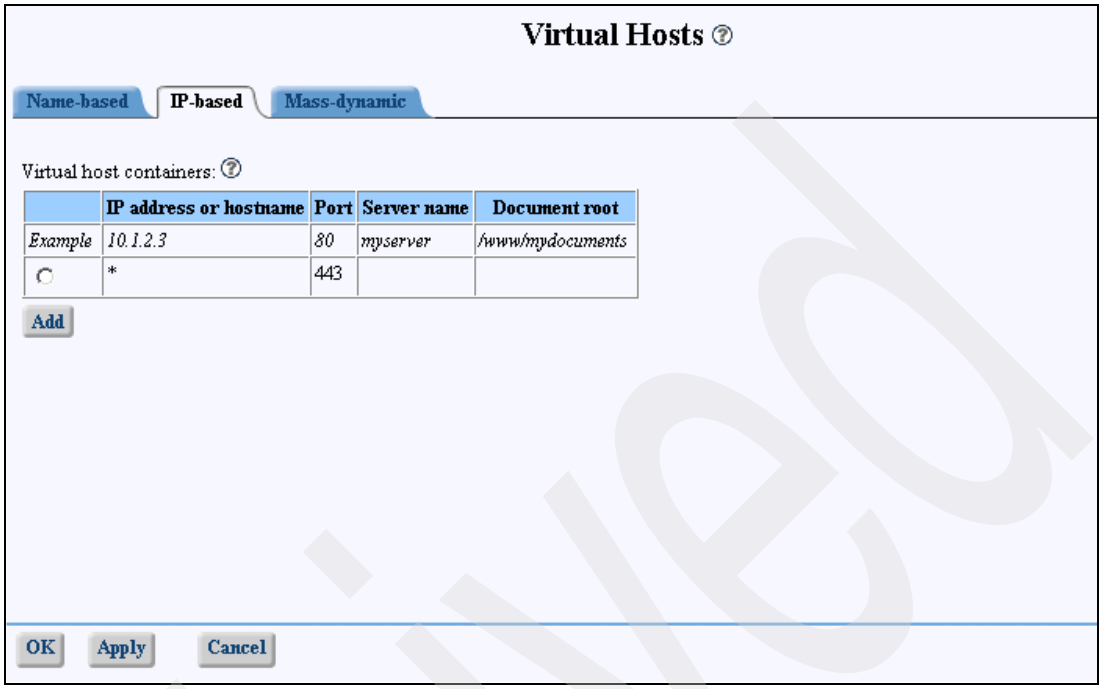


Figure 8-33 Adding a virtual host.

8. Click **Apply**.
9. Your are now required to enable SSL for the new virtual host we created. From the top menu, select the virtual host area as shown in Figure 8-34.

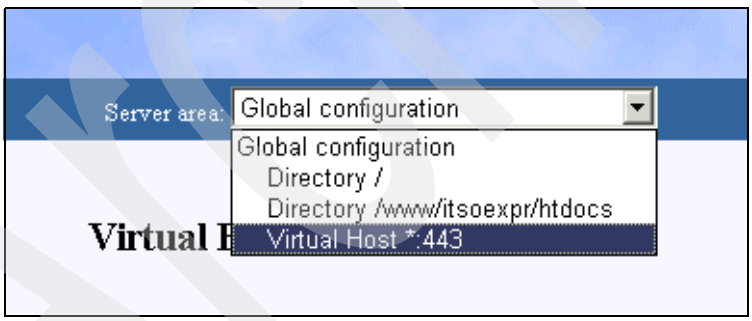


Figure 8-34 Selecting the virtual host area

10. Select **Security** on the task bar. The security settings appear as shown in Figure 8-35.

Figure 8-35 Basic security settings

11. Enable SSL as shown in Figure 8-36. Make sure you select the application name matching your HTTP server as shown.

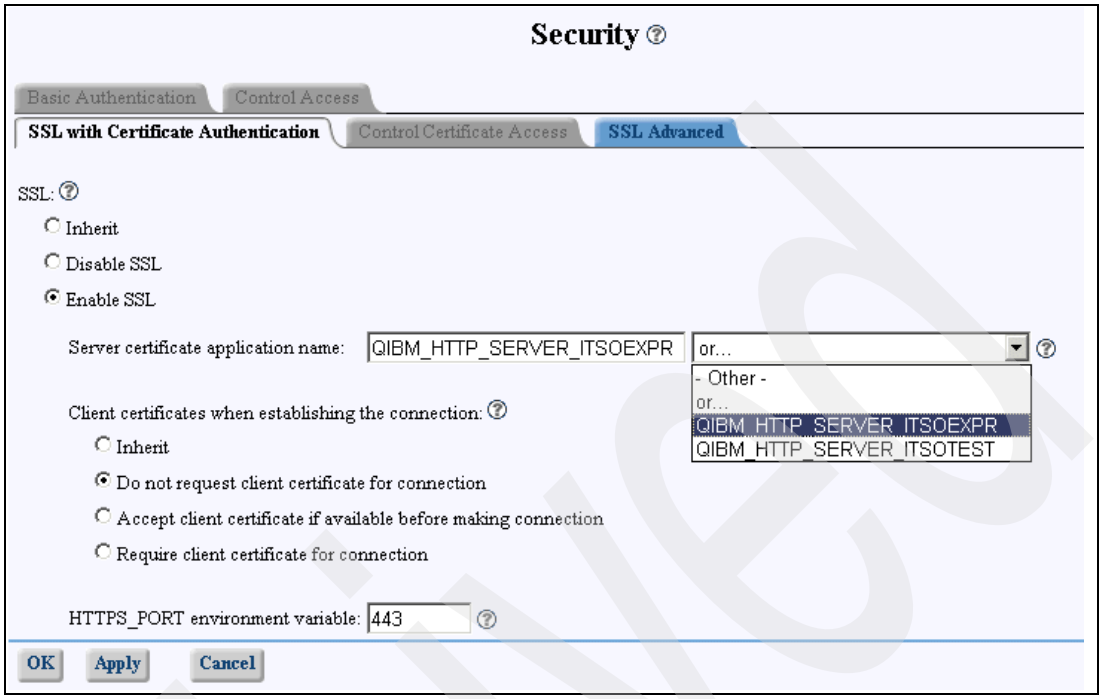


Figure 8-36 Enabling SSL for a virtual host

12. Click **Apply**. You will receive a message saying that the server needs to be restarted. Do *not* do this yet. If you do, the HTTP server will fail to start. We have a few more steps to complete first.

Adding a virtual host for SSL to WebSphere Application Server V5.0

We will now update the virtual host to accepts incoming requests on our SSL port. This is essentially replicating the changes we did to the HTTP server instance.

13. Connect to the WebSphere Application Server Admin client

14. Expand **Environment --> Virtual Host** and click the **Host aliases** link.

15. Add the SSL port as shown and click **Apply** (see Figure 8-37).

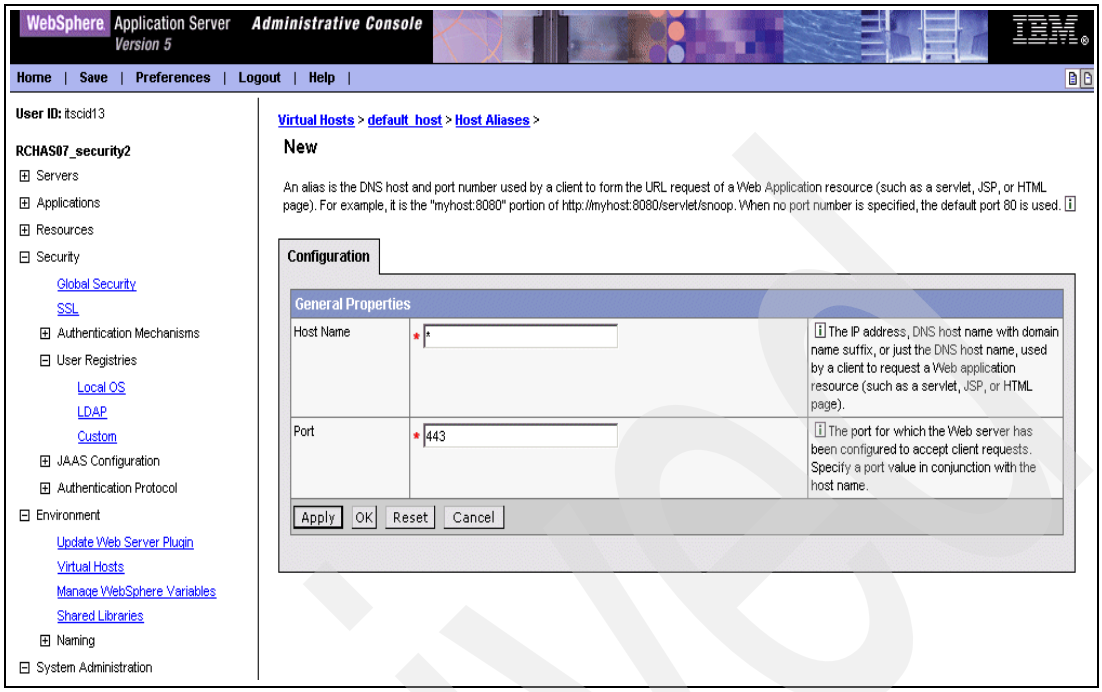


Figure 8-37 Adding the SSL port to the application server virtual host

16. Now stop and start the WebSphere Application Server instance to ensure the changes take effect.
17. Restart your HTTP server instance.

Assign a certificate to the HTTP server instance

Finally we need to assign a certificate to the HTTP server application we selected in "Enabling SSL in your HTTP server instance" on page 376.

1. Start another browser instance and enter the following URL:

   `http://[your iSeries System]:2001`

2. Sign on with your user ID and password:

   Note that the user profile needs to have *ALLOBJ and *SECADM special authorities.

3. Click the **Digital Certificate Manager** link.
4. Click the **Select a Certificate Store** button on the navigation pane. The window appears, as shown in Figure 8-26 on page 371.
5. Select the ***SYSTEM** certificate store and click **Continue**.
6. Enter your password (see step 5 on page 363) and click **Fast Path**.

7. Select **Work with server applications**. A list of available applications are listed as shown in Figure 8-38.

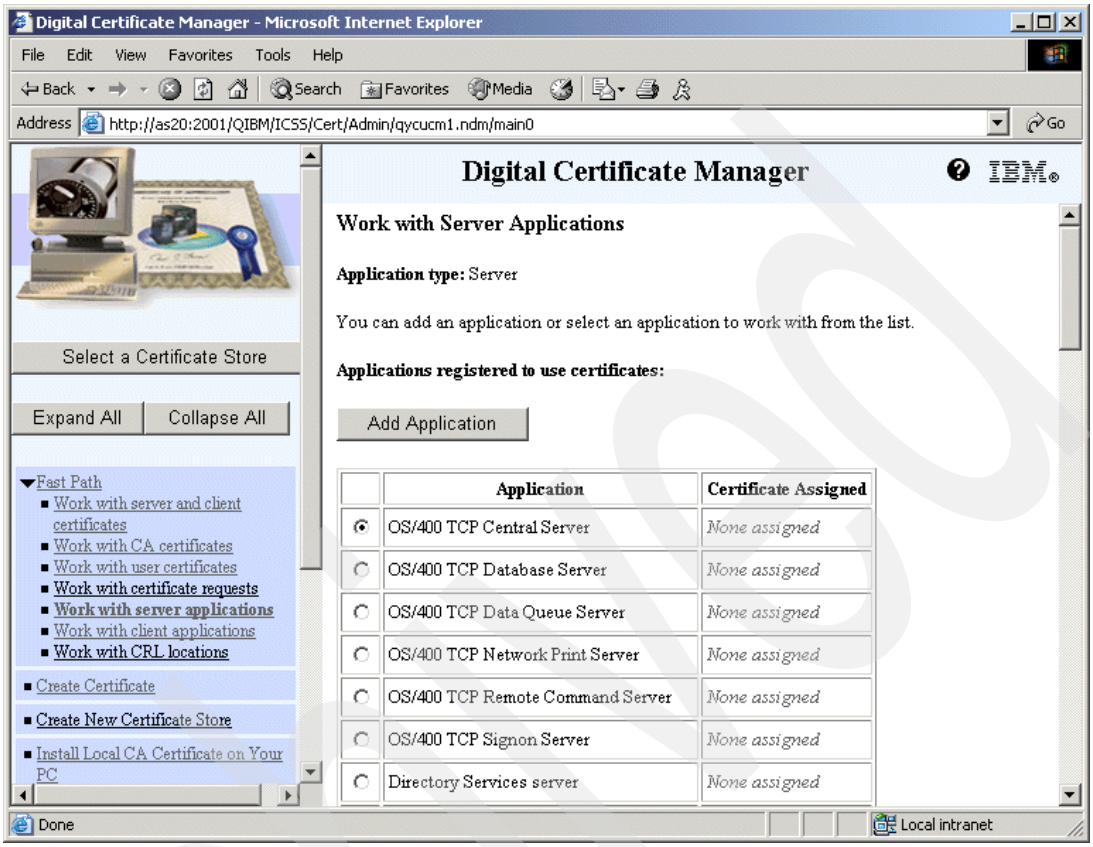


Figure 8-38 Work with server applications

8. Scroll down to locate your application instance. In our case it is name as follows:

   QIBM_HTTP_SERVER_SECURITY

9. Select the application and click the **Work with Application** button. The details of the application appear, as shown in Figure 8-39.

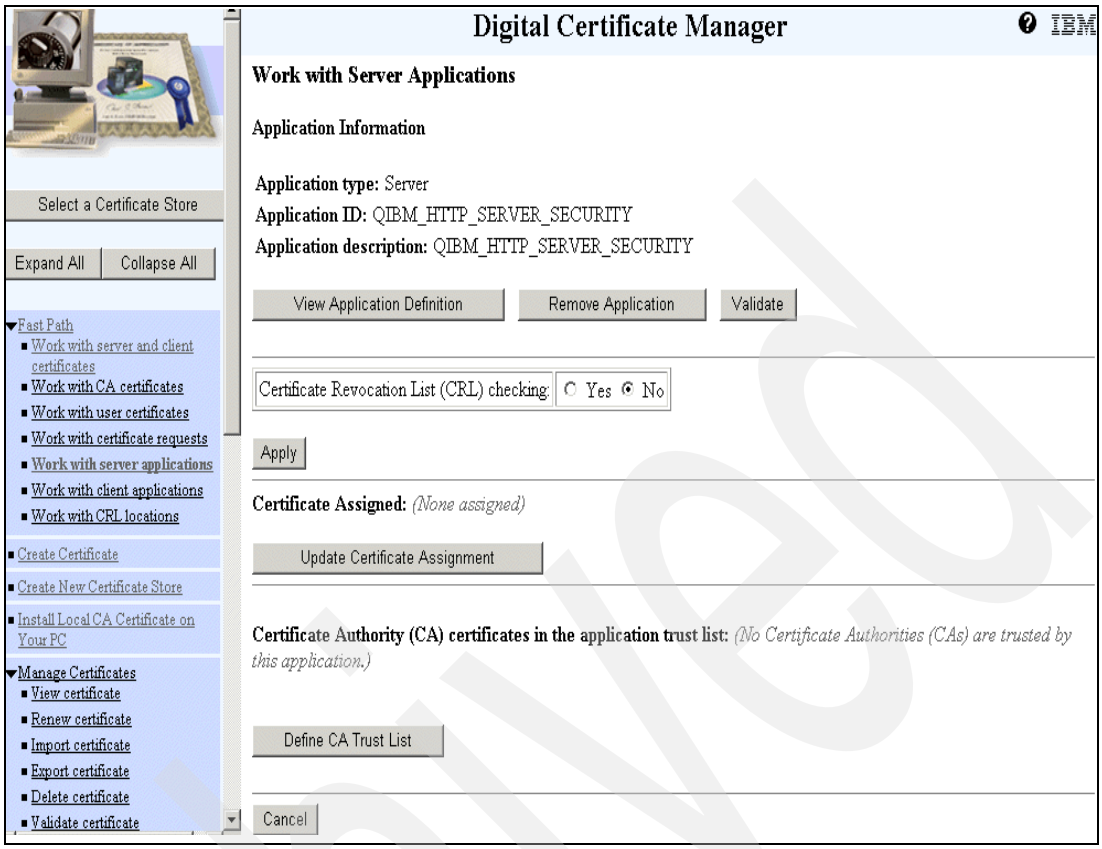


Figure 8-39 Application certificate details.

10. Click on **Update Certificate Assignment** to select a certificate.
11. Select the certificate you created earlier and click **Assign New Certificate**.
12. You will get a confirmation message. Click **Cancel** to exit.

Testing the SSL configuration

We will now verify that the SSL configuration was successful:

1. Return to the browser instance that holds your HTTP server configuration and restart the HTTP server instance.
2. Test that SSL is working using the Snoop servlet. Enter the following (case-sensitive) URL:

   ```
   https://<Your iSeries system>:port/Snoop
   ```

   The browser confirms that you are switching to a secure connection and displays the snoop servlet result.

8.4 Dos and Don’ts while enabling security

These are some important tips about enabling WAS security on the iSeries system:

- At the very least, turn on WebSphere Security for the server.
- Having no security allows anyone to connect to a server and make changes to the production environment.
- Secure the file system holding configuration files and log files.
- Configuration files can hold passwords, and log files may hold confidential “System.out” information.
- Only set Authentication Protocol to “Both” if CSIv2 and SAS are required.
  - Setting to “Both” when only one is needed will cause inappropriate interceptors to be called and result in lower performance.

8.5 More information

This section lists several good sources for further reading about WebSphere Application Server security:

- WebSphere Application Server V5.0 for iSeries InfoCenter at:

  `http://publib.boulder.ibm.com/iseries/v1r1m0/websphere/ic2924/index.htm?info/rzaiz/50/wasindex.htm`

  Expand the **Security** subtree.
- Redbook *IBM WebSphere V5.0 Security WebSphere Handbook Series* at:

  `http://www.redbooks.ibm.com/pubs/pdfs/redbooks/sg246573.pdf`

9

The wsadmin tool

In this chapter we provide an overview of the WSAdmin tool, focusing on the common configurational and operational administrative functionality of the tool. We also provide a basic foundation for building WSAdmin scripts with an introduction to advanced scripting techniques. Finally, we explore migrating **wscp** scripts to **wsadmin** scripts. The topics covered are as follows:

- Overview of wsadmin
- Configuring and launching
- Common configurational and operational administrative tasks with wsadmin, showing useful commands for:
  - AdminConfig object
  - AdminControl object
  - AdminApp object
  - Help object
- Scripting concepts and advanced scripting examples
- Migrating wscp scripts to WSAdmin scripts.

For further information on this topic consult *IBM WebSphere Application Server Version 5.0 Handbook*, SG24-6195-00 and the WAS 5.0 for iSeries Information Center at:

http://publib.boulder.ibm.com/iseries/v1r1m0/websphere/ic2924/index.htm?info/rzaiz/50/wasindex.htm

9.1 Overview of wsadmin

wsadmin is a command line scripting interface that uses the Bean Scripting Framework (BSF). BSF is a set of Java classes that provide access to Java objects via various scripting languages. At this time WAS 5.0 supports only the *Jacl* scripting language.

wsadmin makes Java objects available through language specific interfaces. Scripts use these objects for application management, configuration, operational control, and for communication with MBeans running in WebSphere server processes. MBeans are part of the Java Management Extension (JMX) specification.

To learn more about BSF, go to:

`http://jakarta.apache.org/bsf/manual.html`

To learn more about JMX, go to:

`http://java.sun.com/products/JavaManagement/`

9.2 Configuring and launching wsadmin

This section covers the settings necessary to launch wsadmin, and the different ways to run wsadmin.

9.2.1 Syntax and parameters

Use the following syntax to invoke the wsadmin tool from the QShell command line:

`wsadmin -instance` *instance* [**`-c`** '*command*' | **`-f`** *scriptfile*] [**`-lang`** *lang*] [**`-p`** *propertiesfile*] [**`-profile`** *scriptfile*] [**`-conntype`** *conntype*] [**`-host`** *host*] [**`-port`** *port*]] [*script_parameters*] [**`-h(elp)`**] [**`-wsadmin_classpath`** *wsadmin_classpath*]

Parameters

- **`-instance`**
  This parameter tells wsadmin which instance you are administering. If omitted, it uses the default instance.
- **`-c`**
  Use this parameter to run a single command. wsadmin is instantiated, the command executes, and then wsadmin exits, returning control to the Qshell.
- **`-f`**
  Use this parameter to name a script to run. Like **`-c`**, the **`-f`** parameter transfers control to wsadmin, and then back to Qshell. Both **`-f`** and **`-c`** may not be specified together.
- **`-lang`**
  This parameter overrides the value of **com.ibm.ws.scripting.defaultLang** in the wsadmin.properties file. At this time, only Jacl is supported on the iSeries. You may use this if running a script or profile, and the file extension is not recognized as a supported language. It is a best practice to give Jacl scripts a .jacl extension.
- **`-p`**
  Use this parameter to specify a properties file to use for the new wsadmin session. The values in the named file will override any values in the wsadmin.properties file of the instance or the instance owner. It may be useful to create special properties files for special purposes like enabling trace when debugging.

- **-profile**
  Use this parameter to specify a script to run to initialize the wsadmin environment. This overrides any profiles named in the **com.ibm.ws.scripting.profiles** property. The profile script will run before any command or script if **-c** or **-f** is specified. This may be useful for capturing values you may use again and again in easy to remember variables. This will be discussed in more detail in section 9.4, “Scripting concepts and advanced scripting examples” on page 396.
- **-conntype**
  This parameter overrides the value of **com.ibm.ws.scripting.connectionType**. Select either SOAP or RMI. You will change this to supported protocol of a remote host. You may also select NONE. This will put wsadmin in local mode, and only certain AdminConfig and AdminApp commands will be available.
- **-host**
  This overrides the value of **com.ibm.ws.scripting.host**. Specify the name of remote host where the instance your are to administer resides.

  The following parameters are optional when specifying a remote host:

  -port
  This overrides the value of **com.ibm.ws.scripting.port**. You should name the port for the connection protocol supported by the remote host. If you do not name a port, be sure that the value set in the properties file is compatible with the host.

  -user
  Use this parameter to specify a user ID if authentication is enabled on the named host.

  -password
  Specify the password for the user identified in the **-user** parameter.
- **script_parameters**
  If you specify the -f parameter to run a script, you can include that script's parameters in the wsadmin command. In Jacl the parameters are made available in array **argv** and the number of parameters in **argc**.
- **-help**
  Use this parameter to view help for the wsadmin command.
- **-wsadmin_classpath**
  This optional parameter makes additional classes available to your scripting process. The specified classpath is added to the classloader for the scripting process. This value overrides **com.ibm.ws.scripting.classpath** property.

9.2.2 Launching wsadmin

wsadmin may be run in three different ways:

- Interactively in its own shell.
- Opened for a single command using the **-c** syntax as described in the previous section.
- Opened to run a script using **-f** syntax. This will be described in more detail in 9.4, “Scripting concepts and advanced scripting examples” on page 396.

To run wsadmin in any mode on the iSeries, start by opening a Qshell session and changing directories to the `<WAS_INSTALLATION_ROOT>/bin` directory. Type **wsadmin** at the command line, and the interactive wsadmin shell will start. In Example 9-1, wsadmin is started without passing any parameters. You can see that the instance “default” is used and all other properties are imported from wsadmin.properties.

Example 9-1 Starting the interactive wsadmin shell

```
wsadmin
Could not find parameter -instance, using default "default"
WASX7209I: Connected to process "server1" on node RCHAS07_default using SOAP connector;
WASX7029I: For help, enter: "$Help help"
wsadmin>
```

If you only want to run a single command or script using **-c** or **-f**, control is transferred to wsadmin, the command or script is processed, and then control returns to Qshell.

9.2.3 Configuring wsadmin

The default properties for wsadmin are in the <WAS_ROOT>/properties/wsadmin.properties file. You can change these values by editing this file directly, or they can be overridden at run time. To edit the file directly, use the following command at the iSeries command line:

```
EDTF '/qibm/userdata/WebAS5/<Deployment>/instanceName/properties/wsadmin.properties'
```

In the command above, change *Deployment* to your deployment type (Base or ND) and *instanceName* to the name of your instance. Example 9-2 shows what the wsadmin.properties file looks like.

Example 9-2 wsadmin.properties

```
#---------------------------------------------------------------------
# Properties file for scripting client
#   Base App Server version
#---------------------------------------------------------------------
#
#---------------------------------------------------------------------
# The connectionType determines what connector is used.
# It can be SOAP or RMI.
# The default is SOAP.
#---------------------------------------------------------------------
com.ibm.ws.scripting.connectionType=SOAP
#com.ibm.ws.scripting.connectionType=RMI
#---------------------------------------------------------------------
# The port property determines what port is used when attempting
# a connection.
# The default SOAP port for a single-server installation is 8880
#---------------------------------------------------------------------
com.ibm.ws.scripting.port=8880
#com.ibm.ws.scripting.port=2809

#---------------------------------------------------------------------
# The host property determines what host is used when attempting
# a connection.
# The default value is localhost.
#---------------------------------------------------------------------
#com.ibm.ws.scripting.host=localhost

#---------------------------------------------------------------------
# The defaultLang property determines what scripting language to use.
# Jacl is the sole supported language
#---------------------------------------------------------------------
com.ibm.ws.scripting.defaultLang=jacl

#---------------------------------------------------------------------
# The traceFile property determines where trace and logging
```

```
# output are directed.  If more than one user will be using
# wsadmin simultaneously, different traceFile properties should
# be set in user properties files.
# The default is that all tracing and logging go to the console;
# it is recommended that a value be specified here.
# If the file name contains DBCS characters, use unicode format such as \uxxxx, where xxxx is a number
#-------------------------------------------------------------------------
com.ibm.ws.scripting.traceFile=/QIBM/UserData/WebAS5/Base/default/logs/wsadmin.traceout

#-------------------------------------------------------------------------
# The validationOutput property determines where validation
# reports are directed.  If more than one user will be using
# wsadmin simultaneously, different validationOutput properties should
# be set in user properties files.
# The default is wsadmin.valout in the current directory.
# If the file name contains DBCS characters, use unicode format such as \uxxxx, where xxxx is a number
#-------------------------------------------------------------------------
com.ibm.ws.scripting.validationOutput=/QIBM/UserData/WebAS5/Base/default/logs/wsadmin.valout

#-------------------------------------------------------------------------
# The traceString property governs the trace in effect for
# the scripting client process.
# The default is no tracing.
#-------------------------------------------------------------------------
#com.ibm.ws.scripting.traceString=com.ibm.*=all=enabled

#-------------------------------------------------------------------------
# The profiles property is a list of profiles to be run before
# running user commands, scripts, or an interactive shell.
# securityProcs is included here by default to make security
# configuration easier.
#-------------------------------------------------------------------------
com.ibm.ws.scripting.profiles=/QIBM/ProdData/WebAS5/Base/bin/securityProcs.jacl:/QIBM/ProdData/WebAS5/Base/
bin/LTPA_LDAPSecurityProcs.jacl

#This will not wrap in EDTF

#-------------------------------------------------------------------------
# The emitWarningForCustomSecurityPolicy property controls whether
# message WASX7207W is emitted when custom permissions are found.
# Possible values are: true, false
# The default is "true"
#-------------------------------------------------------------------------
# com.ibm.ws.scripting.emitWarningForCustomSecurityPolicy=true

#-------------------------------------------------------------------------
# The tempdir property determines what directory to use for temporary
# files when installing applications.
# The default is that the JVM decides -- this is java.io.tmpdir
#-------------------------------------------------------------------------
#com.ibm.ws.scripting.tempdir=
#-------------------------------------------------------------------------

# The validationLevel property determines what level of validation to
# use when configuration changes are made from the scripting interface.
# Possible values are: NONE, LOW, MEDIUM, HIGH, HIGHEST
# The default is HIGHEST
#-------------------------------------------------------------------------
#com.ibm.ws.scripting.validationLevel=
```

```
#---------------------------------------------------------------------
# The crossDocumentValidationEnabled property determines whether the validation
# mechanism examines other documents when changes are made to one document.
# Possible values are: true, false
# The default is "true"
#---------------------------------------------------------------------
#com.ibm.ws.scripting.crossDocumentValidationEnabled=

#---------------------------------------------------------------------
# The classpath property is appended to the list of paths to search for
# classes and resources.
# There is no default value.                                          +
#---------------------------------------------------------------------
#com.ibm.ws.scripting.classpath=
```

As you can see in Example 9-2, the wsadmin.properties file is very well documented. For each unique instance, there will be a unique wsadmin.properties file.

9.2.4 Changing wsadmin properties at run time

Editing the wsadmin.properties file will apply changes permanently for the corresponding instance. There may be times, however, when you need to change a property for special task or debugging purposes. Fortunately, wsadmin provides two ways to override properties at run time without editing the wsadmin properties file:

- Create another properties file and name it explicitly at run time.
- Override individual properties in command line arguments

Examples

Example 9-3 and Example 9-4 show how to override properties at runtime:

Example 9-3 Using a custom properties file

```
wsadmin -instance default -p /home/greg/wscustom.properties
WASX7209I: Connected to process "server1" on node RCHASO7_default using SOAP connector;
WASX7029I: For help, enter: “$Help help"
wsadmin>
```

In Example 9-4 the connection type and port are overridden.

Example 9-4 Changing properties at the command line

```
wsadmin -instance default -conntype RMI -port 2809
WASX7209I: Connected to process "server1" on node RCHASO7_default using RMI connector;
WASX7029I: For help, enter: "$Help help"
wsadmin>
```

9.3 Common configurational and operational administrative tasks

This section covers many of the configurational and operational administrative tasks available through wsadmin. It describes how wsadmin may be used to create and modify WebSphere Application server configuration, and how wsadmin may be used to carry out common operational tasks in a WebSphere environment.

Configurational and operational administrative tasks are controlled by four different objects. Each object has a distinct set of operations that are useful for creating, configuring, controlling, modifying, and deploying objects. The four administrative objects are described below:

- **AdminConfig**
  The AdminConfig object is used to access configurational commands to create, configure, modify, and remove objects in the WebSphere architectural structure including nodes, cells, application servers, and more. To see the complete list of types that can be administered with the $AdminConfig object, type the following at the wsadmin command line:

  ```
  $AdminConfig types
  ```

  AdminConfig may operate in two modes: *default* and *local*. In the default mode, AdminConfig communicates with the WebSphere server to accomplish its tasks. In the local mode no server communication occurs. To start wsadmin in local mode, override the **com.ibm.ws.scripting.connectionType** value in the wsadmin.properties file to **NONE** or set the command line parameter **-conntype NONE** as described in 9.2.4, “Changing wsadmin properties at run time” on page 390. A local connection may be desirable when administering only one node in cluster when isolation and encapsulation are concerns.

- **AdminControl**
  The AdminControl object used to control runnable objects. This includes starting, stopping and holding operational processes in WebSphere servers. Admin Control also has utility methods for enabling tracing, reconnecting to a server, and converting data types. To see a list of all AdminControl commands, type the following at a wsadmin command line:

  ```
  $Help AdminControl
  ```

  The AdminControl objects provide support for JMX methods along with its own commands.

- **AdminApp**
  The AdminApp object is used to administer applications. This includes listing, installing, uninstalling, and editing. To see a list of AdminApp commands, enter the following on a wsadmin command line:

  ```
  $Help AdminApp
  ```

  Like the AdminConfig object, the AdminApp object may be operated in local mode by setting the connection type value to none as described in **AdminConfig**.

- **Help**
  The Help object provides useful help for all of the wsadmin objects and wsadmin itself. Help also can give more details about error messages and MBeans registered to the connected server. To see a list of all Help commands, enter the following at the wsadmin command line:

  ```
  $Help help
  ```

  General wsadmin help is available form the command line as well. Type the following for general wsadmin help:

  ```
  $Help wsadmin
  ```

9.3.1 Useful AdminConfig object commands with examples

This section contains AdminConfig command examples. Usually, AdminConfig commands will require three steps when updating an instance configuration.

1. Find the ID of the configuration object to be updated. Use **getid**.
2. Modify the configuration object. This is the same for creating new objects. To create a new Server, its parent object, the Node, needs to be updated.
3. Save the configuration object.

The most useful commands will be discussed here. Be sure to use the online help to see all commands available for the AdminConfig object:

```
$AdminConfig help
```

For commands without examples here, see the WAS 5.0 Infocenter.

getid

This command returns the configId of specific object given its containment. This is the most important configurational command because, without it, no configurational tasks in wsadmin can be accomplished. Therefore, **getid** is covered in greater detail in “Finding objects” on page 397. There you will find explanations of the required arguments and an example.

types

Lists the possible types for configuration. Example 9-5 shows the command and some of the types. There are over 200 types that can be administered.

Example 9-5 Command types with first two listed

```
$AdminConfig types
AdminService
Agent
-- cut here --
```

list

Lists all configuration objects of a given type. This is especially useful when trying to determine what objects can be configured. Example 9-6 shows all of the Server type objects available to be configured for our instance. In this case, there is only one Server object available.

Example 9-6 The list command and results for Server

```
$AdminConfig list Server
server1(cells/RCHASO7_default/nodes/RCHASO7_default/servers/server1:server.xml#Server_1)
wsadmin>
```

parents

Used to show the objects which contain a given type. You will need to know this to create or remove an object. Node is parent to Server, so Node gets updated anytime a Server is created or destroyed. Example 9-7 shows how to get the parent of an object.

Example 9-7 Finding an object’s parent

```
$AdminConfig parents Server
Node
wsadmin>
```

attributes

Used to show the attributes for a given type. This will be helpful when trying discover which attributes you need to change. Example 9-8 lists the attributes for Nodes.

Example 9-8 Node attributes

```
$AdminConfig attributes Node
"discoveryProtocol ENUM(UDP, TCP, MULTICAST)"
"hostName String"
"name String"
"properties Property(TypedProperty)*"
wsadmin>
```

show

Used to show the attributes of a given configuration object. This is like attributes, except it returns the current values of the attributes, as seen in Example 9-9.

Example 9-9 Getting attribute values using show.

```
$AdminConfig show [$AdminConfig getid /Node:RCHAS07_default/]
{hostName RCHAS07}
{name RCHAS07_default}
{properties {}}
wsadmin>
```

defaults

Displays the default values for attributes of a given type.

create

Creates a configuration object, given a type, a parent, and a list of attributes, and optionally an attribute name for the new object. Although it creates a server, create is updating the node to include the server. Example 9-10 shows how the Node ID is obtained, and then the server is created, and finally saved (commands that we type are in bold).

Example 9-10 Creating a server with wsadmin

```
set node [$AdminConfig getid /Node:RCHAS07_default/]
RCHAS07_default(cells/RCHAS07_default/nodes/RCHAS07_default:node.xml#Node_1)
wsadmin>
$AdminConfig create  Server $node { { name tedwig } }
tedwig(cells/RCHAS07_default/nodes/RCHAS07_default/servers/tedwig:server.xml#Server_1)
wsadmin>
$AdminConfig save
```

modify

Changes specified attributes of a given configuration object. Like create, you must get the configId before you can modify the object. Example 9-11 changes the name attribute of our server from `tedwig` to `joshwig`.

Example 9-11 Modifying server name

```
set servername [$AdminConfig getid /Node:RCHAS07_default/Server:tedwig/]
tedwig(cells/RCHAS07_default/nodes/RCHAS07_default/servers/tedwig:server.xml#Server_1)
wsadmin>
$AdminConfig modify $servername { { name joshwig } }
```

remove
Removes the specified configuration object.

convertToCluster
Converts a server to be the first member of a new ServerCluster. Any server added with createClusterMember adopts the first member's configuration

createClusterMember
Creates a new server that is a member of an existing cluster.

save
Commits unsaved changes to the configuration repository. Use this command any time you make a change using wsadmin in the interactive mode.

9.3.2 Useful AdminControl object commands with examples

AdminControl commands, although different in functionality, are similar to AdminConfig in the approach necessary to manipulate objects.

1. Get the object name.
2. Manipulate the object.

completeObjectName and queryNames
Both of these commands are described in greater detail in "Finding objects" on page 397. They are the most important AdminControl commands because with them, names of objects are returned. Without the names, AdminControl has no operational control.

getNode
This returns the node name of the connected server. The same value can be retrieved with completeObjectNames and queryNames, but getNode is a nice little shortcut. The name of the node is used repetitively, especially when creating templates for completeObjectName and queryNames commands. Example 9-12 shows how to get the node name.

Example 9-12 Getting the connected node

```
$AdminControl getNode
RCHAS07_default
wsadmin>
```

getCell
This command is nearly identical to getNode, except it returns the cell name. Use the same syntax as getNode in Example 9-12.

invoke
Invoke a method on the specified MBean. This command really demonstrates the usefulness of JMX. Any registered method on an associated MBean may be invoked with is command. Example 9-13 shows how to invoke the connected server's MBean's `stop` method.

Example 9-13 Invoking server MBean's stop method

```
set servername [$AdminControl completeObjectName node=RCHAS07_default,type=Server,*]
wsadmin>
$AdminControl invoke $servername stop
```

stopServer and startServer

The server may be stopped with the **stopServer** command. This is true for any deployment, Base or ND. The **startServer** command will only work in a managed environment with a nodeAgent and requires a ND installation. In either case, the syntax for the commands is the same. Example 9-14 shows how to stop a server.

Example 9-14 Using the stopServer command

```
$AdminControl stopServer server1 [$AdminControl getNode]
WASX7337I: Invoked stop for server "server1" Waiting for stop completion.
WASX7264I: Stop completed for server "server1" on node "RCHASO7_default"
wsadmin>
```

9.3.3 Useful AdminApp object commands with examples

Many of the most useful application administration functions are listed here.

list

Lists all installed applications. Example 9-15 shows the **list** command.

Example 9-15 Listing applications

```
$AdminApp list
DefaultApplication
adminconsole
ivtApp
wsadmin>
```

listModules

Lists the modules in a specified application. In ND installations, adds the name of the server to list modules for an application on the specified server.

install

Installs an application, given a file name and an option string.

installInteractive

Installs applications and prompts the user for options. Any options defined in the application can be overridden at this time.

uninstall

Uninstalls an application, given an application name and an option string.

editInteractive

Used to edit the properties of an application interactively. You will be prompted for each option of an application.

export

Exports application to a file.

options

Shows the options available, either for a given file, or in general. It is called with the full path to the application. Example 9-16 shows the options command with the full path of application.

Example 9-16 The options command with full path information

```
$AdminApp options /QIBM/userdata/webas5/base/greg/installedapps/RCHASO7_greg/ivtApp.ear
WASX7112I: The following tasks are valid for
"/QIBM/userdata/webas5/base/greg/installedapps/RCHASO7_greg/ivtApp.ear"
MapRolesToUsers
BindJndiForEJBNonMessageBinding
MapEJBRefToEJB
--- cut here ---
```

9.3.4 Useful Help object commands

Help object commands are designed to provide help with wsadmin in general. Context specific help is available for any command by typing *AdminObject* **help** *command name* at the wsadmin command line. *AdminObject* is the wsadmin administrative object, and *command name* is the command. Example 9-17 shows help for install.

Example 9-17 Context sensitive help

```
$AdminApp help install
WASX7096I: Method: install

        Arguments: filename, options

        Description: Installs the application in the file specified by
        "filename" using the options specified by "options." All required
        information must be supplied in the options string; no prompting is
        performed.

        The AdminApp "options" command may be used to get a list of all
        possible options for a given ear file.  The AdminApp "help" command
        may be used to get more information about each particular option.
wsadmin>
```

There is also a $Help object that can be used to get help for any object or command in wsadmin.

9.4 Scripting concepts and advanced scripting examples

This section explains the concepts you should understand to write scripts in the wsadmin environment. At the basic level, you should understand the requirements of wsadmin to complete the desired tasks. For more advanced scripting, you can add structured programming to complete conditional or repetitive tasks.

9.4.1 Basic concepts

Scripts are a set of commands that are read into wsadmin to complete specified tasks. Commands in wsadmin are designed to act on named objects. In the previous examples in 9.3, “Common configurational and operational administrative tasks” on page 390 there is a minimum two step process:

1. Find the name or ID of the object to be acted upon.
2. Act on the object.

Finding objects

To find objects and MBeans, wsadmin provides methods that are dependent upon the type of task. Configuration tasks are accomplished with the AdminConfig object. AdminConfig identifies configurable components with unique IDs. The command to capture an object's identity is:

$AdminConfig getId *containment_path*

Here, *containment_path* is the topological path to the configurable object. To get the JDBC provider, for example, its path would start at the node, then server, and then the JDBC provider like this:

/Node:myNode/Server:s1/JDBCProvider:jdbc1/

In Example 9-18 we get the node ID, so we can add a server.

Example 9-18 Getting a node ID in wsadmin

```
$AdminConfig getid /Node:RCHASO7_default/
RCHASO7_default(cells/RCHASO7_default/nodes/RCHASO7_default:node.xml#Node_1)
wsadmin>
```

Operational tasks are managed by the AdminControl object. To get the operational name of a runnable object, use:

$AdminControl completeObjectName *name,template*
or
$AdminControl queryNames *object_name*

For **completeObjectName** the *name,template* is a single string with name value pairs separated by commas. Example 9-19 shows how the *name,template* argument should be entered.

Example 9-19 Getting the complete name of a server.

```
$AdminControl completeObjectName name=server1,type=Server,*
WebSphere:name=server1,process=server1,platform=common,node=RCHASO7_default,version=5.0,type=Server,mbeanIdentifier=cells/RCHASO7_default/nodes/RCHASO7_default/servers/server1/server.xml#Server_1,cell=RCHASO7_default,processType=UnManagedProcess
wsadmin>
```

The * indicates a wild card in the template. Any name value pair can be listed as part of the argument.

Important: Name value pairs are separated by commas with no spaces. This is important because spaces delimit additional arguments, and since this command only takes one argument, **spaces between pairs will cause errors**.

The more name value pairs you include, the more exact your answer will be. This command only returns the first control object that matches. Example 9-20 shows an example of an imprecise or incomplete template.

Example 9-20 Getting an object name with an incomplete template

```
$AdminControl completeObjectName version=5.0,type=Servlet,*
WASX7026W: String "version=5.0,type=Servlet,*" corresponds to 16 different MBeans;
returning first one.
WebSphere:WebModule=DefaultWebApplication.war,name=Hit Count
Servlet,process=server1,Application=DefaultApplication,platform=common,node=RCHASO7_default
```

```
,J2EEName=DefaultApplication#DefaultWebApplication.war#Hit Count
Servlet,Server=server1,version=5.0,type=Servlet,mbeanIdentifier=DefaultApplication#DefaultW
ebApplication.war#Hit Count Servlet,cell=RCHAS07_default
wsadmin>
```

The **queryNames** command will return all objects that match the *object_name*, which is really the name value pairs. Therefore, the same rules apply to **queryNames** as **completeObjectName**. Example 9-21 shows how to get a list of objects that match an incomplete template.

Example 9-21 Listing all values that match a template

```
$AdminControl queryNames version=5.0,type=Servlet,WebModule=ivt_app.war,*
"WebSphere:WebModule=ivt_app.war,name=JSP 1.2
Processor,process=server1,Application=ivtApp,platform=common,node=RCHAS07_default,J2EEName=
ivtApp#ivt_app.war#JSP 1.2
Processor,Server=server1,version=5.0,type=Servlet,mbeanIdentifier=ivtApp#ivt_app.war#JSP
1.2 Processor,cell=RCHAS07_default"

WebSphere:WebModule=ivt_app.war,name=SimpleFileServlet,process=server1,Application=ivtApp,p
latform=common,node=RCHAS07_default,J2EEName=ivtApp#ivt_app.war#SimpleFileServlet,Server=se
rver1,version=5.0,type=Servlet,mbeanIdentifier=ivtApp#ivt_app.war#SimpleFile
Servlet,cell=RCHAS07_default
wsadmin>
```

You can see that two objects are returned that are type Servlet (`type=Servlet`), are in version 5.0 of WAS (`version=5.0`), and are part of WebModule ivt_app.war (`WebModule=ivt_app.war`).

Acting on objects

Now that you have object names, you can act on objects using commands provided for configurational and operational tasks. However, you may have noticed that the names and IDs for the objects are quite long. It certainly would be difficult to cut and paste object names from an interactive session into a script over and over again. It also would make your script quite ugly and hard to maintain. The solution is to hold values in wsadmin memory and point to them with meaningful names or variables.

Setting variables

You may point to object names by setting variables to the name values. Earlier in Example 9-18 on page 397 we got the ID for our node. To set a variable to hold the node ID, we would do the following:

```
set nodeId [$AdminConfig getid /Node:RCHAS07_default/]
```

Now *nodeId* points to the value of the result of the **getid** command and you can substitute it in any command that takes the node ID as a parameter like this:

```
$AdminConfig show $nodeId
```

Note that *nodeId* is prepended by a '**$**'. This tells wsadmin to use the value pointed to by *nodeId*.

A simple script

Now that you can find and act on objects, you can write a simple script. Example 9-22 shows a simple script to stop a server. This script is saved in the StopServer.jacl file and can be invoked with `-f` option of the wsadmin script (see "Parameters" on page 386).

Example 9-22 StopServer.jacl

```
#StopServer.jacl
#This stops the named server
#
#Get the node we are connected to
set node [$AdminControl getNode]
#Get the complete server object name
set servername [$AdminControl completeObjectName name=server1,type=Server,node=$node,*]
#Write to screen whats happening
puts “Stopping Server $servername”
#Stop the server
$AdminControl invoke $servername stop
#Tell screen its done
puts “$servername stopped”
```

This is a pretty clean, simple script. Setting values makes it easy to write and maintain. Note that comments are indicated by # symbol. Example 9-23 shows what happens when the script is run.

Example 9-23 Results of running StopServer.jacl

```
$
wsadmin -f /pshock/StopServer.jacl
Could not find parameter -instance, using default "default"
WASX7209I: Connected to process “server1” on node RCHAS07_default using SOAP connector;
The type of process is: UnManagedProcess

Stopping Server
WebSphere:name=server1,process=server1,platform=common,node=RCHAS07_default,version=5.0,typ
e=Server,mbeanIdentifier=cells/RCHAS07_default/nodes/RCHAS07_default/servers/server1/server
.xml#Server_1,cell=RCHAS07_default,processType=UnManagedProcess

WebSphere:name=server1,process=server1,platform=common,node=RCHAS07_default,version=5.0,typ
e=Server,mbeanIdentifier=cells/RCHAS07_default/nodes/RCHAS07_default/servers/server1/server
.xml#Server_1,cell=RCHAS07_default,processType=UnManagedProcess stopped
$
```

As you can see, it would be impractical to write scripts if you had to replace each occurrence of `$servername` with `WebSphere:name=server1,process=server1,platform=common,node=RCHAS07_default,version=5.0,type=Server,mbeanIdentifier=cells/RCHAS07_default/nodes/RCHAS07_default/servers/server1/server.xml#Server_1,cell=RCHAS07_default,processType=UnManagedProcess`.

9.4.2 Advanced scripting techniques and examples

This section explores more advanced scripting techniques including structured programming and procedures. It provides some examples of each and points you to more examples on the Web.

Structured programming

Structured programming mechanisms are available to wsadmin through Jacl. Some of the most useful ones are demonstrated here.

if...elseif...else

`if...elseif...else` is used to execute certain commands if a certain condition is met. The construct for conditional execution is as follows:

```
if { boolean expression is true } {
  executable statements -- leave construct
} elseif { another boolean condition is true but not the one preceding this } {
  alternate executable statements -- leave construct
} else {
  some other executable statements
}
```

The `elseif` and `else` portions are not required. Once a condition is satisfied, the script advances to the next line after the last curly brace }. The executable portion must always be enclosed in curly braces { } even if there is only one executable line (see Example 9-24).

Example 9-24 if...else in action

```
set node [$AdminControl getNode]
set servername [$AdminControl completeObjectName name=server1,type=Server,node=$node,*]
set serverstate [$AdminControl getAttribute $servername state]
if { $serverstate == “STARTED” } {
  puts "Server is $serverstate"
} else {
  puts "Server is $serverstate"
}
```

Important: The white space on each side of the curly braces is required. Without it, your script will fail like this:

```
WASX7017E: Exception received while running file "/pshock/checkServer.jacl";
exception information: com.ibm.bsf.BSFException: error while eval'ing Jacl
expression: invalid command name "if{STARTED=="STARTED"}"
```

Also, note that the *else* line must start with the closing curly brace of the *if* executable section. Again, the script fails when *else* starts a line:

```
WASX7017E: Exception received while running file "/pshock/checkServer.jacl";
exception information: com.ibm.bsf.BSFException: error while eval'ing Jacl
expression: invalid command name "else"
```

Whitespace and newline are very important. Whitespace delimits unique “words”, and newline delimits each executable line.

foreach

`foreach` provides a means of executing the same commands for each occurrence of something (an object or value). The construct for **`foreach`** is like this:

```
foreach item in a list {
do this
and this
}
```

You may wish to stop all servers in a node for example. Instead of doing so one server at a time, Example 9-25 shows how to do just that using **`foreach`**.

Example 9-25 StopServers.jacl using foreach

```
set node [$AdminControl getNode]
set servername [$AdminControl queryNames type=Server,node=$node,*]
puts ""
foreach item $servername {
puts "Stopping $item\n"
```

```
$AdminControl invoke $item stop
}
```

Note that **queryNames** returns a list of all servers that match the template, and **foreach** iterates through the list and points a new variable at the current item so that it can be referenced during each iteration.

for

When an action needs to be taken only a certain number of times, a for control loop may be useful. The paradigm for **for** is as follows:

```
for { aValue } { aValue meets condition } { change aValue } {
  execute statements
}
```

The loop iterates until the condition is false. Usually, a counter is set and then incremented after each iteration. The loop continues until the counter exceeds some threshold. Perhaps don't want to stop all servers, only certain number, and it is not important which ones. You can do this with a for loop as seen in Example 9-26.

Example 9-26 Using a for loop

```
set node [$AdminControl getNode]
set servername [$AdminControl queryNames type=Server,node=$node,*]
puts ""
set response ""
#We know we have only 2 servers
  while { ($response!= "0" && $response != "1" && $response != "2") } {
    puts "How many servers do you want to stop?    "
    puts "Enter a number 0 to 2."
    set response [gets stdin]
  }
puts "Stopping $response servers in node $node ."
for { set i 0 } { $i < $response } {incr i } {
  set item [lindex $servername $i]
  puts "Stopping $item\n"
  $AdminControl invoke $item stop
}
puts "Done"
```

while

while is another looping mechanism that executes while a condition is true. It is a hybrid of **if** and **foreach**. It built like this:

```
while { boolean expression is true } {
  do this
  and this too
}
```

This can be very dangerous. If the condition never changes to false, the loop will run infinitely. Example 9-27 shows how to use a while loop. It also show how to prompt a user for an interactive session.

Example 9-27 Using while and prompting users

```
set node [$AdminControl getNode]
set servername [$AdminControl queryNames type=Server,node=$node,*]
puts ""
set response ""
  while { ($response != "1" && $response != "2") } {
```

```
    puts "Are you sure you want to stop all servers?"
    puts "Enter 1 for yes and 2 for no."
    set response [gets stdin]
  }
if { $response == "1" } {
  puts "Stopping all servers in node $node ."
  foreach item $servername {
    puts "Stopping $item\n"
    $AdminControl invoke $item stop
  }
} else {
    puts "Not stopping any servers"
}
```

Procedures

Procedures are similar to Java methods or C++ functions. They exist in the current wsadmin environment only, and must be defined for any new wsadmin session. They can be declared in both profile and runtime scripts. If instantiated in a profile script, they are available to both an interactive session and command script.

Declaring and creating a procedure

A procedure is declared like this:

```
proc myProcedure { optionalParm1 optionalParm2 ... optionalParmN } {
  global myGlobalVarialble
  localVariables
  executable statements
  return value
}
```

- **proc** is a keyword declaring the procedure.
- myProcedure is your arbitrary procedure name.
- Optional parameters are optional because a procedure doesn't need to have parms to exist. However, if parms are declared, they should be there upon execution.
- **global** is a keyword that makes a variable that exists outside the procedure available to the procedure. Any changes made to the global variable inside the procedure will be available outside the procedure.
- Any local variables declared inside the procedure only exist inside the procedure and are not available to the wsadmin environment.
- The executable statements portion is just like any Jacl script.
- **return** is a keyword that returns a value to the procedure caller.

Executing procedures

A procedure may be executed any time after it is declared. Depending on the environment in which it is declared, a procedure may be executed a number of different ways.

- Interactively. A procedure may be called from the wsadmin command line if it is declared in a profile script that is executed at wsadmin start up without -c or -f values.
- Inside the script that declared it. A procedure is always available to the script that declared it. If a procedure returns no values, it can be executed like any command:

myProcedure parm1 parm2

If a procedure returns a value, then it can be executed in an expression:

set done myProcedure
or
if { myProcedure } { ...

- From another script. If wsadmin is launched using a profile script, then its procedures are available to any script executed inside that wsadmin instance.

In Example 9-28, a previous Example 9-26 on page 401 is transformed to a procedure.

Example 9-28 Transform a script to a procedure

```
proc stopSomeServers { } {
# Make AdminControl available to this procedure
  global AdminControl
  set node [$AdminControl getNode]
  set servername [$AdminControl queryNames type=Server,node=$node,*]
  puts ""
  set response ""
  while { ($response!= "0" && $response != "1" && $response != "2") } {
    puts "How many servers do you want to stop?    "
    puts "Enter a number 0 to 2."
    set response [gets stdin]
  }

  puts "Stopping $response servers in node $node ."
  for { set i 0 } { $i < $response } {incr i } {
    set item [lindex $servername $i]
    puts "Stopping $item\n"
    $AdminControl invoke $item stop
  }
  puts "Done"
# End proc
}
```

After starting wsadmin with **-profile** and this script, `stopSomeServers` becomes executable in the interactive environment. Example 9-29 shows how wsadmin was launched and how the procedure was executed.

Example 9-29 Executing a profile script to make a procedure available to the interactive environment

```
wsadmin -profile /pshock/stopServersPrf.jacl
Could not find parameter -instance, using default "default"
WASX7209I: Connected to process "server1" on node RCHAS07_default using SOAP connector;
The type of process is: UnManagedProcess
WASX7029I: For help, enter: "$Help help"
wsadmin>
stopSomeServers

How many servers do you want to stop?
Enter a number 0 to 2.
0
Stopping 0 servers in node RCHAS07_default .
Done

wsadmin>
```

More scripts

There are another twelve advanced scripts to study on your iSeries. They can be found at:

- /QIBM/ProdData/WebAS5/Base/bin/LTPA_LDAPSecurityProcs.jacl
- /QIBM/ProdData/WebAS5/Base/bin/securityProcs.jacl
- /QIBM/ProdData/WebAS5/Base/util/mdbSamplesSetup.jacl
- /QIBM/ProdData/WebAS5/Base/samples/bin/MessageDrivenBeans/startMDBSamples.jacl
- /QIBM/ProdData/WebAS5/Base/samples/bin/MessageDrivenBeans/stopMDBSamples.jacl
- /QIBM/ProdData/WebAS5/Base/samples/bin/create_jms_resources.jacl
- /QIBM/ProdData/WebAS5/Base/samples/bin/create_ds.jacl
- /QIBM/ProdData/WebAS5/Base/samples/bin/create_xads.jacl
- /QIBM/ProdData/WebAS5/Base/samples/bin/mail_url_resources.jacl
- /QIBM/ProdData/WebAS5/Base/samples/bin/setupjms.jacl

In these scripts you will find examples of procedures and structured programming. You will also find more detailed examples of configurational and operational administrative tasks. These scripts are very well documented, and will greatly enhance your wsadmin education.

Other IBM Redbooks

SG24-6573-00, *IBM WebsSphere V5.0 Security Handbook* -- Appendix D *Using wsadmin scripting for security configuration* has sample security configuration scripts and information for making wsadmin more secure.

SG24-6195-00, *IBM WebSphere Application Server V5.0 Handbook* -- Chapter 24 *Command line administration and scripting* has more examples and more analysis of how wsadmin really works.

WebSphere Application Server InfoCenter

The information center has many practical examples at:

```
http://publib7b.boulder.ibm.com/webapp/wasinfo1/index.jsp?deployment=ApplicationServer
```

Once there, expand the left hand navigation to **System administration->Scripting**

The WebSphere Application Server V5.0 for iSeries InfoCenter at:

```
http://publib.boulder.ibm.com/iseries/v1r1m0/websphere/ic2924/index.htm?info/rzaiz/50/
admin/wsaexample.htm
```

WebSphere Developer Domain Web site

This site has multiple sample scripts that can be useful while learning wsadmin:

```
http://www7b.boulder.ibm.com/wsdd/library/samples/SampleScripts.html
```

9.5 Migration from wscp to wsadmin

Unfortunately there is no direct migration path from wscp to wsadmin. In this section, however, we will attempt point you in the right direction so that you can begin converting some of your more basic wscp scripts to wsadmin scripts. This section provides mapping tables of similar commands and objects along with references to example replacement commands.

9.5.1 Mapping wscp command objects to wsadmin configuration types

This section focuses on mapping WAS V4.0 wscp command objects to WAS V5.0 wsadmin configuration types. Because there are fewer WAS V3.5 wscp command objects, and because they map easily to WAS V4.0 wscp objects, V3.5 wscp object migration will not be examined here. Use Table 9-1 to map the wscp commands and objects to their wsadmin counterparts.

Table 9-1 WAS V4.0 wscp command objects mapped to WAS V5.0 configuration types

wscp	wsadmin
ApplicationServer	Server
Context	Not Applicable
DataSource	WAS40DataSource, DataSource
Domain	Not Applicable
EnterpriseApp	ApplicationDeployment and/or AdminApp
GenericServer	Server
J2CConnectionFactory	J2CConnectionFactory
J2CResourceAdapter	J2CResourceAdapter
JDBCDriver	JDBCDriver
JMSDestination	JMSDestination
JMSConnectionFactory	JMSConnectionFactory
JMSProvide	JMSProvider
Mail Session	MailSession
Module	ModuleDeployment and/or Admin App
Node	Node
ServerGroup	ServerCluster
URL	URL
URLProvider	URLProvider
VirtualHost	VirtualHost

9.5.2 Mapping wscp operational commands to wsadmin operational commands

The operational commands in wscp are created for each object they control. Operational commands in wsadmin, however, follow a framework that allows them to control any registered object, not just a subset of them. Nearly all of wscp operational commands map to AdminControl object commands, except for security controls. Use Table 9-2 to map operational commands.

Table 9-2 wscp control commands and wsadmin control objects and commands

wscp 4.0 action	wsadmin 5.0 object and command	wsadmin 5.0 Mbean	MBean wsadmin 5.0 Operation
server start	AdminControl		
server stop	AdminControl		
servergroup start	AdminControl invoke	Cluster	start
servergroup stop	AdminControl invoke	Cluster	stop
application start	AdminControl invoke	ApplicationManager	startApplication
application stop	AdminControl invoke	ApplicationManager	stopApplication
node stop	AdminControl	nodeagent	stopNode
check run-time attribue	AdminControl getAttribute	mbean	attribute
check run-time attributes	AdminControl getAttributes	mbean	list of attributes
regenPluginCfg	AdminControl invoke	PluginCfgGenerator	generate
testConnection	AdminControl testConnection		
enable security	securityon command in securityProcs.jacl		
disable security	securityon command in securityProcs.jacl		

9.5.3 More detailed information

More detailed information regarding migrating your wscp scripts to wsadmin can be found at the WAS V5.0 for iSeries Information Center at:

http://publib.boulder.ibm.com/iseries/v1r1m0/websphere/ic2924/index.htm?info/rzaiz/50/wasindex.htm

Once there, expand the left-hand navigation to **Administration>Administrative tools> The wsadmin administrative tool**

Click on the link **Migrate from wscp to wsadmin**.

Other IBM Redbooks

Read the following IBM Redbook for more information on migration issues: *Migrating to WebSphere V5.0: An End-to-End Migration Guide*, SG24-6910-00 at:

http://www.redbooks.ibm.com/pubs/pdfs/redbooks/sg246910.pdf

10

Backup and recovery

In this chapter we present various topics in WebSphere Application Server maintenance. We describe how to back up your WebSphere information, and provide a detailed description of recovery procedures for the WebSphere product.

As a maintenance task, we include a check list for the program temporary fix (PTF) application criteria.

Finally, we describe how you can replicate a WebSphere instance to other systems using the backup that you already have from the original system.

10.1 Backing up WebSphere Application Server

Like any other software running on the iSeries system, WebSphere Application Server has different kinds of objects that allow it to work well. Sometimes, when a problem occurs on the system or in the product itself, you may need to have a backup for your application and your databases. The iSeries system has many options that make possible a 99.94% average system reliability[1], but problems happen anyway, so you should be prepared to recover your information.

You should be sure that your iSeries backup strategy includes all things related to the WebSphere applications; not only the product, but also all objects related to your application. This topic includes specifications about how you have to back up the different kinds of objects associated with WebSphere Application Server.

The WebSphere Application Server uses many iSeries resources that you should consider adding to your backup and recovery procedures. Save and restore of WebSphere Application Server resources uses the same iSeries commands as for other iSeries resources. For more detailed information about backup on iSeries, see the *Backup and Recovery Guide* by selecting the **System Management** link in the iSeries Information Center at:

`http://publib.boulder.ibm.com/pubs/html/as400/infocenter.html`

You need to consider certain aspects of for WebSphere Application Server in your backup and recovery processes. The following areas are described in detail in the rest of this chapter:

- Licensed product
- Administrative configuration
- Servlets
- JavaServer Pages (JSP) files
- HTTP Configuration
- Enterprise Beans
- Security
- Java Message Service (JMS) resources

WebSphere Application Server resources can be saved while the WebSphere Application Server environment is active. When backing up database data, you may have to shut down some or all services if a snapshot cannot be obtained. This would occur if there are requests which obtain locks or have open transactions against the database being saved.

In a distributed environment, you may need to consider how to get a consistent backup across several systems. If the data on some systems is not closely related to data on others, you may be able to back up each system in isolation. However, if you need to take a snapshot across several systems simultaneously, you may need to stop activity on all systems while the snapshot is taken.

How often you back up resources depends largely on when or how often you expect them to change. Use the following categories to determine how you should fit WebSphere resources into your backup plan:

- **WebSphere Application Server environment configuration:** This category covers the resources that define your WebSphere Application Server operating environment. Once you have done initial setup, this information should change very infrequently. You might back up this information only when you change these settings, and not include these resources in regularly scheduled backups. Items included in this group are:

[1] According to the Gartner Group studies. More Information can be found at `http://www.gartner.com`

  - The administrative configuration
  - HTTP configuration
  - Servlet configurations files
  - Security properties files
- **WebSphere Application Server applications:** This category covers the applications you run using WebSphere Application Server. You should back them up in the same way you back up other applications on your system. You could back up these resources every time you add or change an application, or include these resources in a regularly scheduled backup. Items included in this group are:
  - The administrative configuration
  - Servlet source and class files
  - JSP source and generated class files
  - Deployed EJB jar files
- **WebSphere Application Server application data:** This category covers the data stores used by your WebSphere Application server applications. Unless your applications serve only static information, these resources are usually quite dynamic. You should back up these items the same way you back up other business data on your system. These resources are suited for inclusion in a regularly scheduled backup. Items included in this group are:
  - Servlet user profile data
  - Enterprise bean
  - Database data

10.1.1 Saving and restoring licensed products

WebSphere Application Server on iSeries uses some additional licensed products — for example, Java — therefore you should consider these additional products when you define your backup plan.

You should save these products after applying the PTFs related to these products. While you save these products, there should not be any instances or WebSphere jobs running.

These are the products that you must consider in your backup plan:

- OS/400 Version 5 Release 1 (V5R1) or Version 5 Release 2 (V5R2)
- IBM Developer kit for Java (5722JV1) Version 1.3 (option 5).
- OS/400 Qshell (572SS1 option 30).
- HTTP Server:
  - IBM HTTP Server (powered by Apache)(5722DG1).
  - Lotus Domino for AS/400 R5 (5769LNT)
  - Lotus Domino for iSeries 6.0 (5733-LD6)
- WebSphere Application Server (5733WS5):
  - Option *Base,1,2 and 3(samples gallery)
  - Option 5, if using WebSphere Application Server Network Deployment on iSeries.

Note: If you are only using WebSphere Application Server Network Deployment in your iSeries, you must consider save Option *Base and Option 5 of WebSphere Application Server product (5733WS5).

Note: End all application server instances that use JMS and the QMQM subsystem before saving WebSphere MQ.

If you use a WebSphere embedded JMS provider, you should save the following products:

- WebSphere MQ V5.3 for iSeries (5724B41).
- WebSphere MQ classes for Java and JMS V5.3 for iSeries (5769C34).

For all of these products, you should use the SAVLICPGM command of iSeries. For more information about backup on iSeries, see *Backup and Recovery Guide* by selecting the **System Management** link in the iSeries Information Center.

We now provide an example of the SAVLICPGM command; you can choose what products you want to save, options, language, and so on. Figure 10-1 shows the command that saves both libraries and IFS directories associated with each licensed product.

```
                      Save Licensed Program (SAVLICPGM)

Type choices, press Enter.

Product  . . . . . . . . . . . . > 5733WS5       Character value
Device . . . . . . . . . . . . . > *SAVF         Name, *SAVF
              + for more values
Optional part to be saved  . . .   *BASE        *BASE, 1, 2, 3, 4, 5, 6, 7...
Release  . . . . . . . . . . . .   *ONLY        Character value, *ONLY
Language for licensed program  . > *ALL         Character value, *PRIMARY...
Object type  . . . . . . . . . .   *ALL         *ALL, *PGM, *LNG
Check signature  . . . . . . . .   *SIGNED      *SIGNED, *ALL, *NONE
Save file  . . . . . . . . . . .                Name
  Library  . . . . . . . . . . .     *LIBL      Name, *LIBL, *CURLIB
Target release . . . . . . . . .   *CURRENT     *CURRENT, *PRV, V4R5M0...
Clear  . . . . . . . . . . . . .   *NONE        *NONE, *ALL, *AFTER, *REPLACE
Data compression . . . . . . . .   *DEV         *DEV, *NO, *YES
License acceptance required  . .   *NO          *NO, *YES


                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 10-1 SAVLICPGM example

The restore procedure should be done using the command RSTLICPGM (see Figure 10-2). You must be sure that all licensed products are restored before you try to use the WebSphere Application Server product.

Important: The following OS/400 commands need to be performed if the WebSphere MQ products are restored *after* WAS:

- `QSYS/CHGUSRPRF USRPRF(QMQM) STATUS(*ENABLED) PWDEXPITV(*NOMAX)`

  This command changes the password expiration interval for the QMQM user profile to *NOMAX (it doesn't expire).

- `QSYS/CHGUSRPRF USRPRF(QMQMADM) STATUS(*ENABLED) PWDEXPITV(*NOMAX)`

  This command changes the password expiration interval for the QMQMADM user profile to *NOMAX (it doesn't expire).

- `QMQM/CHGMQMCAP CAPUNITS(*YES)`

  This command is used to indicate that the sufficient license units have been purchased for this installation of WebSphere MQ.

The advantage of saving WebSphere Application Server as a licensed product, is that you can avoid the time for PTF installation, because this backup will contain all PTFs applied until the backup. Of course, if you are going to use this backup to restore WebSphere Application Server on another system, you must be sure that all the other products like Java, QShell, etc., and/or PTFs required, are installed on the system — not only for WebSphere Application Server, but also for the other products.

```
                    Restore Licensed Program (RSTLICPGM)

Type choices, press Enter.

Product  . . . . . . . . . . . . > 5733WS5       Character value
Device . . . . . . . . . . . . . > *SAVF         Name, *SAVF
             + for more values
Optional part to be restored . .   *BASE         *BASE, 1, 2, 3, 4, 5, 6, 7...
Type of object to be restored  .   *ALL          *ALL, *PGM, *LNG
Language for licensed program  .   *PRIMARY      Character value, *PRIMARY...
Output . . . . . . . . . . . . .   *NONE         *NONE, *PRINT
Release  . . . . . . . . . . . .   *FIRST        Character value, *FIRST
Replace release  . . . . . . . .   *ONLY         Character value, *ONLY, *NO
Save file  . . . . . . . . . . .                 Name
  Library  . . . . . . . . . . .     *LIBL       Name, *LIBL, *CURLIB

                                                                      More...
F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
F13=How to use this display        F24=More keys
```

Figure 10-2 RSTLICPGM example

10.1.2 Saving and restoring your administrative configuration

The administrative configuration contains vital information about your WebSphere Application Server setup, and should be backed up. Most of the configuration for your WebSphere Application Server instance resides in the config directory structure. In addition, the properties directory also contains several important configuration files. For more information on the configuration content stored in the properties directory, see 2.18, "IFS directory structure" on page 46.

Note: The commands below may have been wrapped for display purposes. Enter each as a single command.

Save and restore these properties files using the Save (SAV) and Restore (RST) commands, for example:

```
SAV DEV('/QSYS.lib/wsalib.lib/wsasavf.file')
OBJ(('/QIBM/UserData/WebAS5/Base/<instanceName>/properties/*')
('/QIBM/UserData/WebAS5/Base/<instanceName>/config/*'))
```

Here, `<instanceName>` is the name of your instance. The default instance name is `default`.

Note: You must restore these directories to the same directory name and path that you back them up.

You must be sure that all restored files are owned by the user profile used to start all the jobs related to the WebSphere application Server instance. By default, `QEJBSVR` user profile must be the owner of all instances files.

Tip: You can use the **chgown** command from Qshell in order to change the owner for an specific directory including all files in it. For example:

```
chgown -R QEJBSVR /QIBM/UserData/WebAS5/Base/default
```

This command makes QEJBSVR the owner for all files inside the default directory.

10.1.3 Saving and restoring Enterprise Applications

Application code and configuration (such as bindings) is located by default in the `/QIBM/UserData/WebAS5/Base/<instanceName>/installedApps` directory. By saving this directory, you save your installed applications, including HTML, servlets, JavaServer Pages (JSP) files, and enterprise beans. Normally, each application is located in a separate subdirectory, so you can choose to save all applications or a subset.

This command will save all installed applications:

```
SAV DEV('/QSYS.lib/wsalib.lib/wsasavf.file')
OBJ(('/QIBM/UserData/WebAS5/Base/<instanceName>/installedApps'))
```

This command will save the sampleApp application only:

```
SAV DEV('/QSYS.lib/wsalib.lib/wsasavf.file')
OBJ(('/QIBM/UserData/WebAS5/Base/<instanceName>/installedApps/cellName/sampleApp.ear'))
```

If you have located utility or general purpose classes in other directories, be sure to include those locations in your backup plan as well; for example:

```
/QIBM/UserData/WebAS5/Base/<instanceName>/lib/app
/QIBM/UserData/WebAS5/Base/<instanceName>/lib/ext
```

Restore should be done by this command:

```
RST DEV('/QSYS.lib/wsalib.lib/wsasavf.file')
OBJ(('/QIBM/UserData/WebAS5/Base/<instanceName>/installedApps/cellName/sampleApp.ear'))
```

Here, `<instanceName>` is the name for the instance. The default instance name is `default`.

10.1.4 Saving and restoring security information

At the time you define your backup strategy, you must consider the security information that will be needed. In this section, we discuss the main elements of WAS security.

Users

When using local OS security, back up your OS/400 user profiles, using normal OS/400 save procedures for user profiles. For more information, go to the *Backup and Recovery Guide* by selecting the **System Management** link in the iSeries Information Center.

For information on the Directory Services Product (LDAP server), see the iSeries Information Center.

For information on Domino, see the Domino Reference Library at:

http://doc.notes.net/domino_notes/5.0/as400/as400hlp.nsf

Security properties files

Security settings are saved in several properties files. By default, these files are located in /QIBM/Userdata/WebAS5/Base/<instanceName>/properties where <instanceName> is the name of your instance. The default instance name is default. If you have defined additional WebSphere instances, you will have additional properties files located in the directories for those instances. Use this command to save the security property files:

```
SAV DEV('/QSYS.lib/wsalib.lib/wsasavf.file')
OBJ(('/QIBM/UserData/WebAS5/Base/<instanceName>/properties/sas*')
('/QIBM/UserData/WebAS5/Base/<instanceName>/config/cells/<NodeName>_<InstanceName>/sec
urity.xml) ('/QIBM/UserData/WebAS5/Base/<instanceName>/properties/soap.client.props'))
```

Here, <instanceName> is the name for the instance and <Node> is the name of your node.

Note: Security property files can be saved while WebSphere is running.

Key files

Key files should also be saved. They contain certificates used by the WebSphere Application Server security infrastructure and also for HTTPS transport between servers. Save all files in the WAS_INSTANCE_ROOT/etc directory. Key files are contained in the WAS_INSTANCE_ROOT/etc directory, but may be created and stored in other directories by administrators.

Validation lists

Passwords are stored as encrypted data in validation list objects when the OS/400 password encoding algorithm is used.

The default validation list is /QSYS.LIB/QUSRSYS.LIB/EJSADMIN.VLDL, but you can change it in the administrative console by specifying it as a system property for the application server.

Save and restore validation list objects using the Save Object (SAVOBJ) and Restore Object (RSTOBJ) commands, for example:

```
SAVOBJ OBJ(EJSADMIN) LIB(QUSRSYS) DEV(*SAVF) SAVF(WSALIB/WSASAVF)
RSTOBJ OBJ(EJSADMIN) SAVLIB(QUSRSYS) DEV(*SAVF) OBJTYPE(*VLDL) SAVF(WSALIB/WSASAVF)
```

10.1.5 Saving and restoring Java Message Service (JMS) resources

You must consider many elements related to Java Message Service (JMS), besides the Enterprise application files, to define your backup strategy. We now describe what resources should be considered for your backup plan.

WebSphere MQ V5.3 for iSeries queue manager data

WebSphere MQ V5.3 for iSeries queue managers are used by the WebSphere embedded JMS provider and should be saved as part of your backup plan. This is how to back up your WebSphere MQ V5.3 for iSeries queue manager data:

1. Ensure that your application server instances that use JMS are running.
2. Display the current list of WebSphere MQ V5.3 for iSeries queue managers using the following command:

   ```
   WRKMQM
   ```

   Note that the names of all queue managers that are prefixed by “WAS_”.
3. Execute the following command for each of these queue managers:

   ```
   RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME('QueueManagerName')
   ```

   Here, `QueueManagerName` is the name of the queue manager(s) noted in the previous step and is enclosed in single quotation marks. For example:

   ```
   RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME('WAS_MYISERIES_myAppServer_myAppServer')
   ```

   Note: The QueueManagerName is case-sensitive.
4. End all application server instances that use JMS.
5. Locate the libraries that are used to store the WebSphereMQ V5.3 for iSeries queue manager data by executing the following command:

   ```
   WRKOBJ QMWAS_* *LIB
   ```

   Note the names of all the libraries that are displayed by this command.
6. Save each of the libraries reported by the command executed in the previous step:

   ```
   SAVLIB LIB(<LibName>) DEV(*SAVF) SAVF(WSALIB/SaveFileName)
   ```

   Here, `<LibName>` is the name of the WebSphere MQ V5.3 for iSeries queue manager data library and `SaveFileName` is the name of the save file to use to store the saved library.
7. Locate the integrated file system directories that are used to store the WebSphere MQ V5.3 for iSeries queue manager data by executing the following command:

   ```
   WRKLNK OBJ('/QIBM/UserData/mqm/qmgrs/*')
   ```

   Note the names of all the integrated file system directories that are prefixed with “WAS_”. To display the complete directory name, position the cursor over each directory and press F22.
8. Save the integrated file system directories for the WebSphere MQ V5.3 for iSeries queue managers using the following command:

   ```
   SAV DEV('/QSYS.lib/wsalib.lib/SaveFileName.file')
   OBJ(('/QIBM/UserData/mqm/qmgrs/DirectoryName'))
   ```

   Here, `SaveFileName` is the name of the save file that is to contain the saved data and `DirectoryName` is the name of each queue manager integrated file system directory identified in the previous step. For example:

   ```
   SAV DEV('/QSYS.lib/wsalib.lib/wsqmsavf.file')
   OBJ(('/QIBM/UserData/mqm/qmgrs/WAS_MYISERIES_myAppServer_myAppServer'))
   ```

10.1.6 Saving and restoring the HTTP configuration

Changes to the HTTP configuration are often made to enable WebSphere Application Server to serve servlets and JSP requests, and to enable WebSphere Application Server security. You should consider saving your HTTP configuration as a part of your WebSphere Application Server backup and recovery. If your HTTP is installed on a different platform (xSeries™, pSeries and so on) than iSeries, refer to the Backup and Recovery guide of that platform to define your backup plan.

Note: The following information applies to IBM HTTP Server for iSeries (powered by Apache). If you are using Lotus Domino HTTP Server, see the Notes.net Documentation Library at:

http://www.notes.notesua.nsf?OpenDatabase

HTTP server instances for IBM HTTP Server for iSeries (powered by Apache) are members of the QATMHINSTC file in the library QUSRSYS. An example save command for this file could be as follows:

```
SAVOBJ OBJ(QATMHINSTC) LIB(QUSRSYS) DEV(*SAVF) OBJTYPE(*FILE) SAVF(WSALIB/WSASAVF)
```

The HTTP configurations for IBM HTTP Server for iSeries (powered by Apache) are stored in the integrated file system in a subdirectory, chosen when the configuration was created. The recommended location is within the WebSphere instance directory. You can determine this file location by inspecting HTTP server instance member in the QATMHINSTC file in library QUSRSYS. An example save command for this file could be as follows:

```
SAV DEV('/QSYS.lib/wsalib.lib/wsasavf.file')
OBJ(('/QIBM/UserData/WebAS5/Base/<instanceName>/apache/conf')
('/QIBM/UserData/WebAS5/Base/<instanceName>/htdocs'))
```

Here, <instanceName> is the name of your instance. The default instance name is default.

Restore should be done by these commands (the examples of the commands correspond to the examples of the SAV commands):

```
RSTOBJ OBJ(QATMHINSTC) SAVLIB(QUSRSYS) DEV(*SAVF) OBJTYPE(*FILE) SAVF(WSALIB/WSASAVF)
```

For the HTTP configurations for BM HTTP Server for iSeries(Powered by Apache):

```
RST DEV('/QSYS.lib/wsalib.lib/wsasavf.file')
OBJ(('/QIBM/UserData/WebAS5/Base/<instanceName>/apache/conf')
('/QIBM/UserData/WebAS5/Base/<instanceName>/htdocs'))
```

Here, <instanceName> is the name of your instance.

10.2 Exporting your .EAR file

Note: The .EAR files contain all resources for your application, like JSP, Servlets, EJB and so on. However, they don't contain anything about configuration, so you should back up your licensed products, WebSphere Application Server configuration information, HTTP configuration, Java Message Services, and data.

Another option that you can use to back up your application files is to export your enterprise application already installed. Using this option, you can save the .EAR file exported into a tape and/or you can keep this file into an IFS directory of the iSeries. The exporting process can be done while the enterprise application is running. You can do that from the administrative console, by following these steps:

1. Start the administrative console.
2. Expand Applications in the topology view and click **Enterprise Application** (see Figure 10-3).

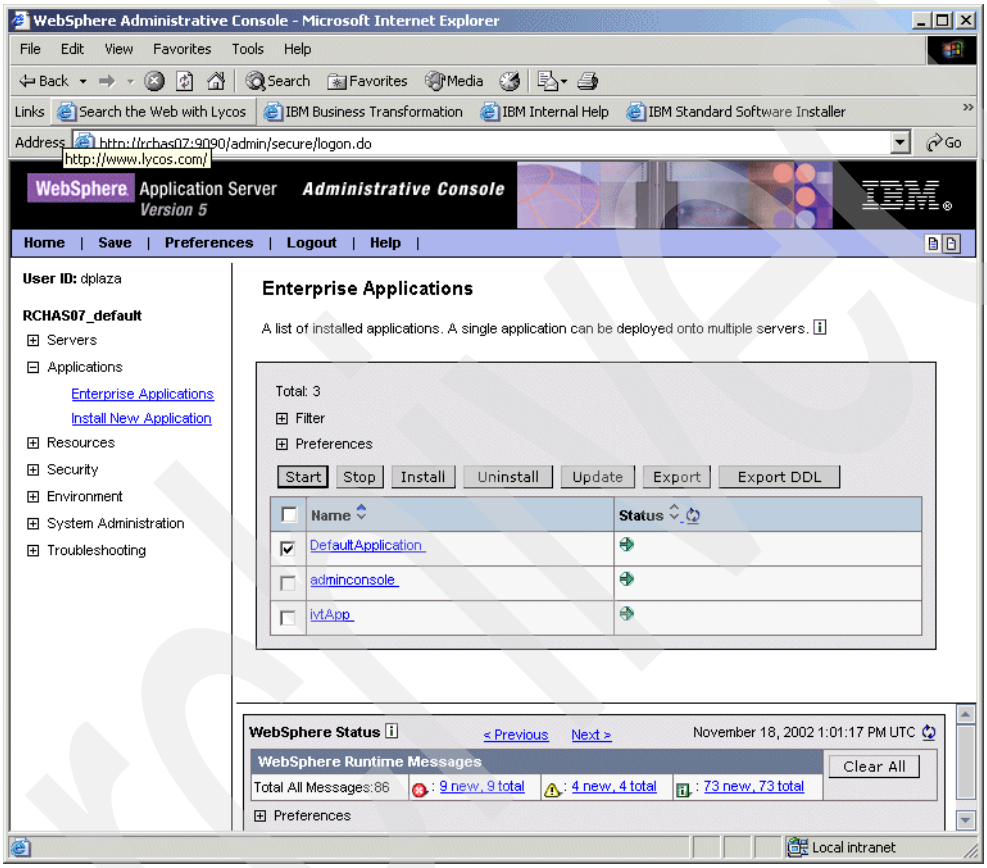


Figure 10-3 Administrative Console example

3. Select the application that you want to export, and click **Export**.
4. Click the Enterprise application that you want to export (see Figure 10-4).

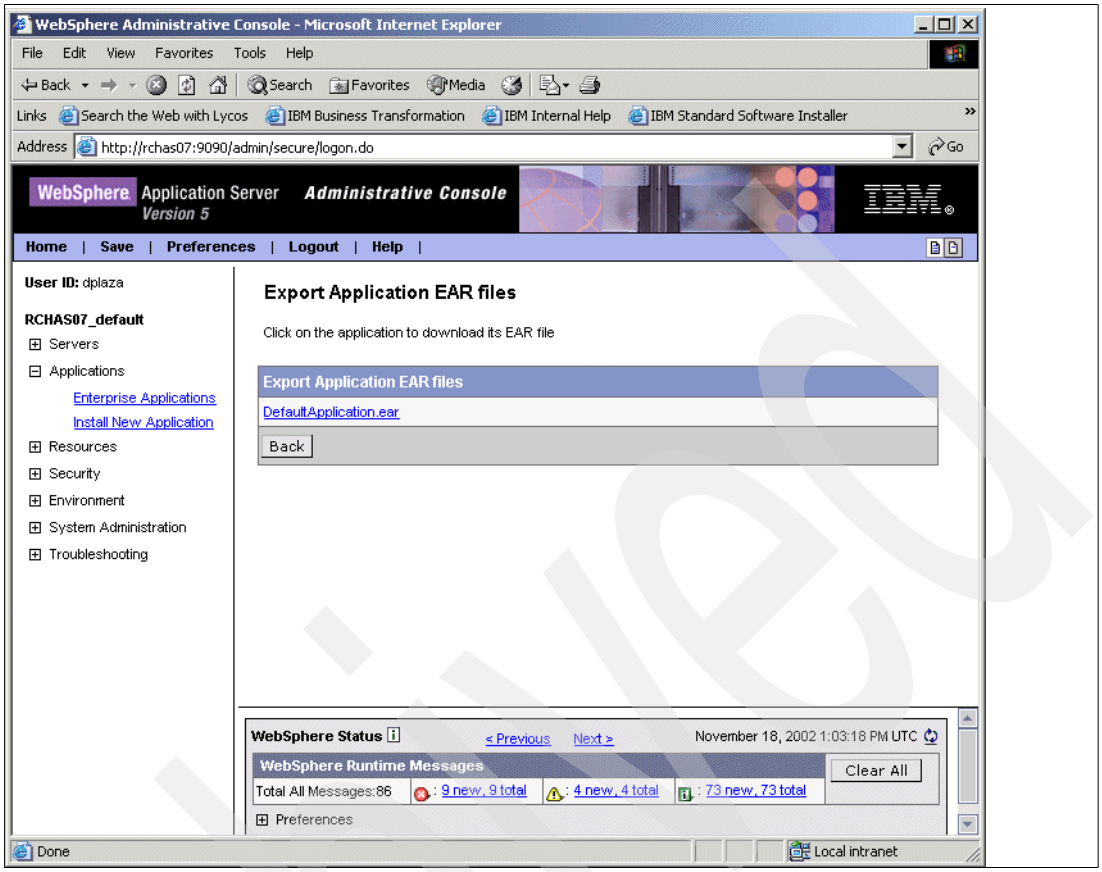


Figure 10-4 Example to select an Enterprise Application

5. Select **Save** in the dialog box and choose the location for the EAR file (see Figure 10-5).

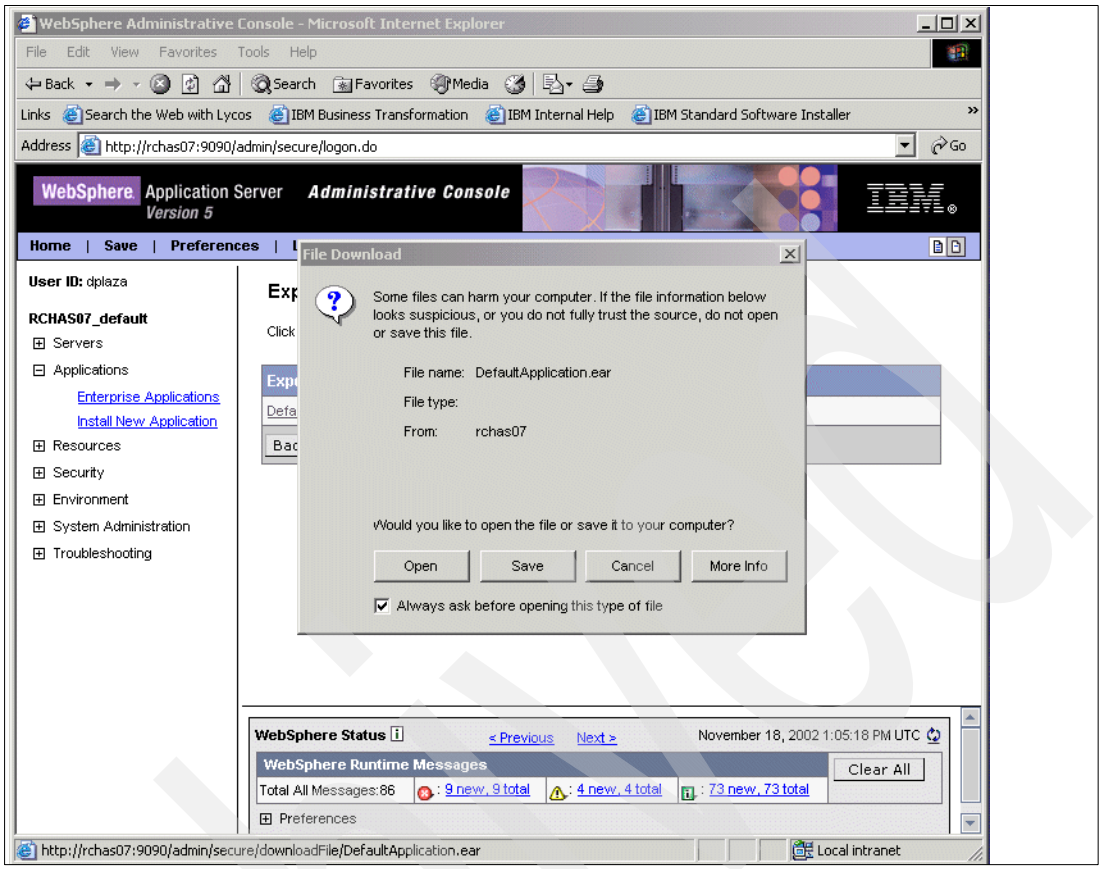


Figure 10-5 Example to export your Enterprise Application

You can use this EAR file to install this application on another system or for recovery proposes. For more information about installing application, refer to 6.13, “Working with Web applications in a WAS environment” on page 187.

10.3 Backing up additional directories

According to the directory structure for each instance, you can choose what kind of additional backup you need to do in order to keep all your Web application information and configurations safe. To see an example of the directory structure for the `default` instance, see Figure 10-6.

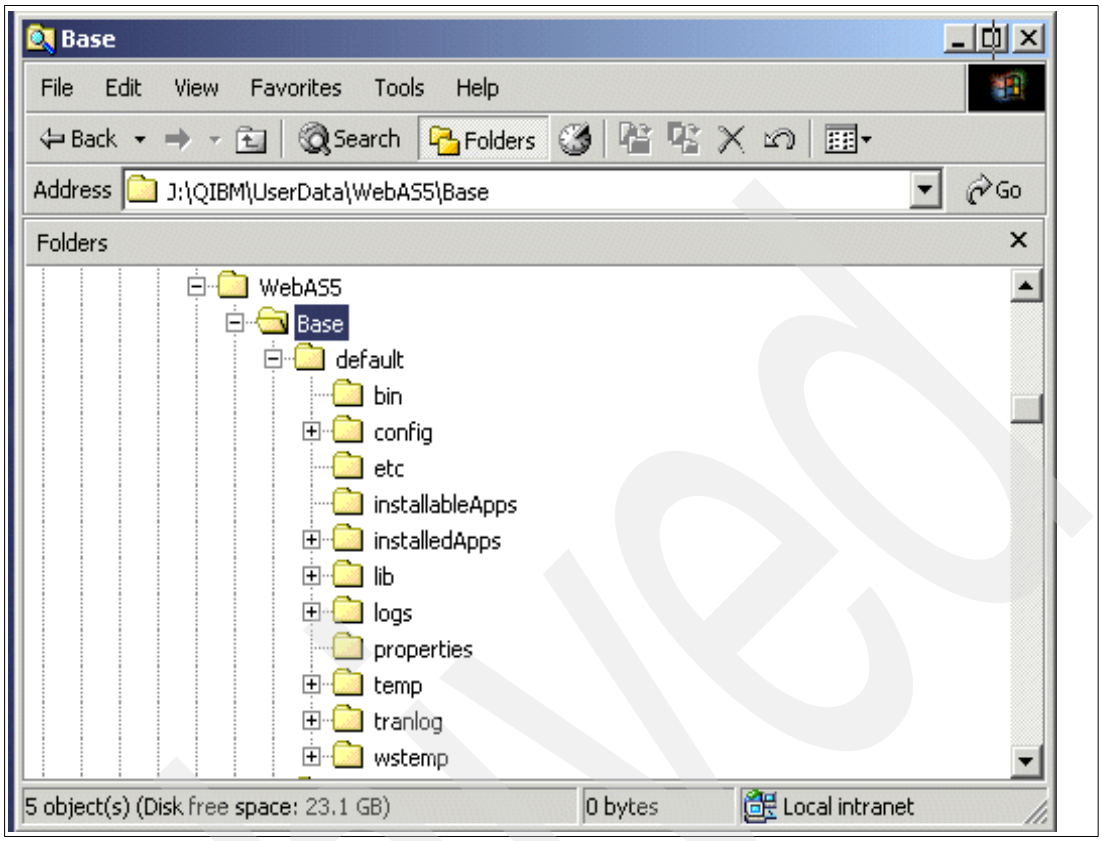


Figure 10-6 Example of the directory structure for a default instance

Each directory contains a different type of information of your application, and we have already covered backup recommendations for almost all of these directories. However, there are other directories that are not covered by the normal backup process, such as wstemp, logs, and temp directories. These directories contain information about:

- **wstemp:** Session information about each user using admin console
- **logs:** Default directory for logs like activity log, SystemErr.log, SystemOut.log, and so on
- **temp:** Used to store the generated files (.class, .java, and optionally, .dat and so on)

If you want to see additional information about the directory structure of WebSphere Application Server, refer to 2.18, “IFS directory structure” on page 46.

Depending of the nature of your installation, you can decide not to back up these directories.

10.4 Backup strategy recommendations

The backup strategy for WebSphere Application Server on your system will depend on the frequency of changes that you make to your configuration and your application. Here, we offer you some recommendations about backup strategy:

- Minimally, you should consider backing up the licensed products each time you update them by application of PTFs for these licensed products.
- Your WebSphere Application Server application should not change very frequently, so you could consider backing up this information only when you change its settings and/or its resources.

- Your WebSphere Application Server environment configuration should not change very frequently, so you might want to consider not including this information in your regular backup strategy.
- Usually your database information changes by every transaction made by your WebSphere Application Server applications, so you must consider a daily backup strategy for this kind of information.

10.5 Recovery from a failure

The recovery procedures you must follow depend on the kind of failure. We describe the general steps needed to recover your WebSphere configuration, assuming that you have already recovered microcode and operating system from your iSeries. If you need additional information about recovery on iSeries, refer to *Backup and Recovery,* SC41-5304.

10.5.1 Recovering WebSphere Application Server licensed product

In order to recover WebSphere Application Server you have to restore it from the last licensed product backup. Refer to 10.1.1, “Saving and restoring licensed products” on page 409 to review the products that you need to restore.

If you don’t have any of these backups available, you must install WebSphere Application Server from the distribution media, and you also must install all licensed products required to WebSphere Application Server. Refer to 2.10, “Installing WebSphere Application Server 5.0 for iSeries” on page 25. Likewise, you must install the last group PTFs for WebSphere Application Server. For more information about the PTFs required for WebSphere Application Server, refer to 10.6, “PTF maintenance” on page 425.

10.5.2 Recovering WebSphere Application Server application

When you have restored WebSphere Application Server licensed product, you are able to install your application. You must restore the administrative configuration for your application (refer to 10.1.2, “Saving and restoring your administrative configuration” on page 411). After that, you should restore all code associated with your application, such as Servlets, Java Server Pages (JSP), HTTP Configuration, Security, and so on. To know what kind of backup you need, refer to 10.1, “Backing up WebSphere Application Server” on page 408.

Note: If you are not recovering your application as a part of a total system failure, you must define all authorities on the IFS directories before starting your WebSphere Application Server Application.

10.5.3 Recovering WebSphere Application Server application data

According to the nature of the failure, you must consider how to restore the databases associated with your WebSphere Application Server Application. This backup must be a part of your normal backup strategy for iSeries.

10.5.4 Restoring WebSphere Application Server instance to another system

Occasionally, you might need to restore all instance information from one system to another system, which doesn’t have the same node name. For example, if something happened on your iSeries where you have WebSphere Application Server working, and you don’t have any high availability implementation for WebSphere, you might need to set up your Web application on another system. There are several steps you must follow to make this work.

Prerequisites

You must have a current backup from the IFS directory of your WebSphere instance. If you are using the iSeries HTTP Server, you must have also a backup for the HTTP instance. To see additional information about backing up your WebSphere Application Server instance, refer to 10.1, “Backing up WebSphere Application Server” on page 408. For example you can use a backup for your WebSphere Application Server instance like this:

```
SAV DEV('/QIBM/UserData/WebAS5/Base/<instanceName>/*')
OBJ(('/QSYS.LIB/WSALIB.LIB./WSASAVF.FILE')) SUBTREE(*ALL)
```

Here, <instanceName> is the name of your WebSphere Application Server instance. For backing up the HTTP instances you can use this command:

```
SAVOBJ OBJ(QATMHINSTC) LIB(QUSRSYS) DEV(*SAVF) SAVF(WASLIB/HTTPINFO)
```

Restoring the instance information

You must follow these steps in order to restore your instance information (all examples correspond to the examples from 10.1.2, “Saving and restoring your administrative configuration” on page 411):

1. Create a directory with the same name as your original WebSphere Application Server instance inside /QIBM/UserData/WebAS5/Base
2. Restore your backup instance to another system. In our case we use this command
   ```
   RST DEV('/QIBM/UserData/WebAS5/Base/<instanceName>/*')
   OBJ(('/QSYS.LIB/WSALIB.LIB./WSASAVF.FILE')) SUBTREE(*ALL)
   ```

Here, <instanceName> is the name of your WebSphere Application Server instance. Make sure that all restored files are owned by the user profile used to start all the jobs related to the WebSphere application Server instance. By default, QEJBSVR user profile must be the owner of all instances files.

Tip: You can use the **chgown** command from Qshell in order to change the owner for a specific directory, including all files in it. For example:

```
chgown -R QEJBSVR /QIBM/UserData/WebAS5/Base/default
```

This command makes QEJBSVR the owner for all files inside the default directory.

3. To restore the HTTP instance:
   a. RSTOBJ OBJ(QATMHINSTC) SAVLIB(QUSRSYS) DEV(*SAVF) SAVF(WASLIB/HTTPINFO) RSTLIB(WASLIB)
   b. CPYF FROMFILE(WSALIB/QATMHINSTC) TOFILE(QUSRSYS/QATMHINSTC) FROMMBR(<HttpInstance>) TOMBR(<HttpInstance>) MBROPT(*ADD)
      Here, <HttpInstance> is the name of your HTTP instance.
   c. Restore the HTTP configurations for BM HTTP Server for iSeries(Powered by Apache):
      ```
      RST DEV('/QSYS.lib/wsalib.lib/wsasavf.file')
      OBJ(('/QIBM/UserData/WebAS5/Base/<instanceName>/apache/conf')
      ('/QIBM/UserData/WebAS5/Base/<instanceName>/htdocs'))
      ```
      where <instanceName> is the name of your instance.
      The default instance name is default.

Checking file authority

You must give the appropriate authority for all files of your HTTP configuration to the QTMHHTTP user profile. You can use the CHGAUT command to grant authority for these files. For example, you must be sure that QTMHHTTP user profile has *RX data authority for the following directory, because inside this directory will be the `plugin-cfg.xml` file:

```
CHGAUT OBJ('/qibm/userdata/webas5/base/<instanceName>/config/cells') USER(QTMHHTTP)
DTAAUT(*RX)
```

Note: If you are using Domino with WebServer, you must give the same authorities, but for the QNOTES user profile.

We show a list of the main files that you must check in order to allow QTMHHTTP user profile to use them:

Note: On our example, we defined the server root for the HTTP instance as /QIBM/UserData/WebAS5/Base/<instanceName>/<HttpInstance>.

You must check your own server root for your HTTP instance.

- CHGAUT OBJ('/QIBM/UserData/WebAs5/Base/<instanceName>/<HttpInstance>') USER(QTMHHTTP) DTAAUT(*RX)
- CHGAUT OBJ('/QIBM/UserData/WebAS5/Base/<instanceName>/<HttpInstance>/*') USER(QTMHHTTP) DTAAUT(*RX)
- CHGAUT OBJ('/QIBM/UserData/WebAS5/Base/<instanceName>/<HttpInstance>/conf') USER(QTMHHTTP) DTAAUT(*RX)
- CHGAUT OBJ('/QIBM/UserData/WebAS5/Base/<instanceName>/<HttpInstance>/conf/*') USER(QTMHHTTP) DTAAUT(*RX)
- CHGAUT OBJ('/QIBM/UserData/WebAS5/Base/<instanceName>/<HttpInstance>/htdocs/*') USER(QTMHHTTP) DTAAUT(*RX)
- CHGAUT OBJ('/QIBM/UserData/WebAS5/Base/<instanceName>/<HttpInstance>/logs/*') USER(QTMHHTTP) DTAAUT(*RX)
- CHGAUT OBJ('/QIBM/UserData/WebAS5/Base/<instanceName>/config') USER(QTMHHTTP) DTAAUT(*X)
- CHGAUT OBJ('/QIBM/UserData/WebAS5/Base/<instanceName>/config/cells') USER(QTMHHTTP) DTAAUT(*RX)
- CHGAUT OBJ('/QIBM/UserData/WebAS5/Base/<instanceName>/config/cells/plugin-cfg.xml') USER(QTMHHTTP) DTAAUT(*RX)
- CHGAUT OBJ('/QIBM/UserData/WebAS5/Base/<instanceName>/etc') USER(QTMHHTTP) DTAAUT(*RX)
- CHGAUT OBJ('/QIBM/UserData/WebAS5/Base/<instanceName>/logs') USER(QTMHHTTP) DTAAUT (*RWX)
- CHGAUT OBJ('/QIBM/UserData/WebAS5/Base/<instanceName>/logs/http_plugin.log') USER(QTMHHTTP) DTAAUT(*RWX)

Here, `<instanceName>` is the name of your WebSphere Application Server instance, and `<HttpInstance>` is the name of your HTTP instance.

Changing directory names

If you are restoring the WebSphere Application Server instance information to another system which has a different node name, you must do the following steps to change the directory names related to the node name.

Inside the directory structure for each WebSphere Application Server instance, there are some directories with the name of the node. You must change this name to the node name for the system where you are restoring all instance information.On our example we saved the instance information from the RCHAS07 node, and we restore the instance information to the RCHASM27 node. The name structure for these directories is: `<Node_Name>_<InstanceName>`. Here, `<NodeName>` is the node name and `<InstanceName>` is the WebSphere Application Server instance name:

1. Map a network drive on your Windows workstation to the directory /QIBM/UserData/WebAS5/Base/<InstanceName> on your iSeries.
2. Use the search tool from Windows Explorer to search for the `<Node_Name>_<InstanceName>` of your old instance, for example, `RCHAS07_diana`. See Figure 10-7 for an example.

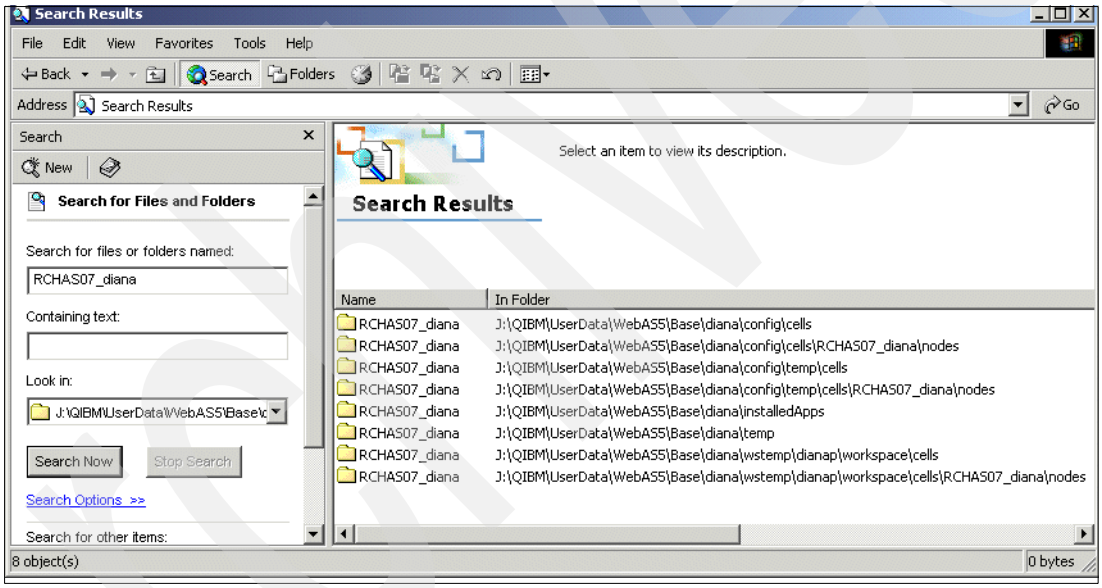


Figure 10-7 Example of search results

3. Rename all of these directories to the appropriate name of the new node. You can use Windows Explorer utilities to do that. See Figure 10-8 for an example.

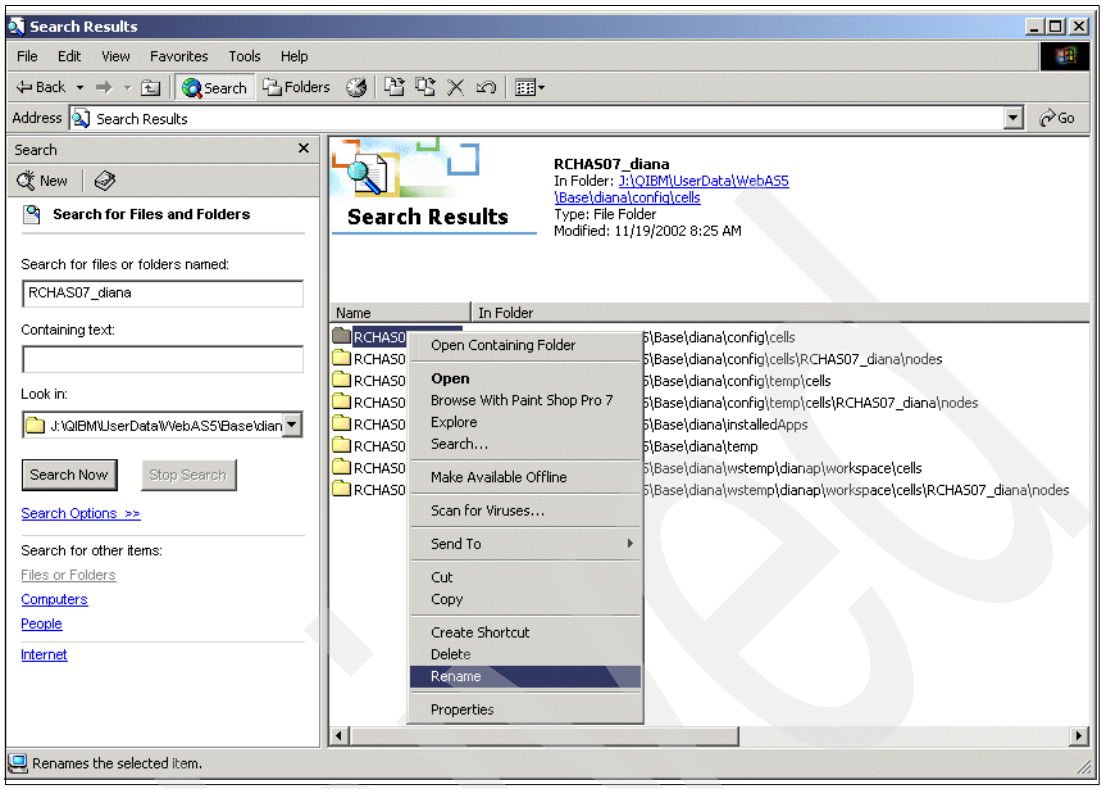


Figure 10-8 Example to rename a directory

Changing Node Name in the instance files

If you are restoring the WebSphere Application Server instance information to another system which has a different node name, you must do the following steps to change this information in the WebSphere Application Server instance files.

Inside some instance files, there is the node name. You must change these node names to the node name of the system where you are restoring all instance information. In our example we saved the instance information from the RCHAS07 node, and we restored the instance information to the RCHASM27 node. The name structure is: `<Node_Name>_<InstanceName>`. Here, `<NodeName>` is the node name and `<InstanceName>` is the WebSphere Application Server instance name:

1. Map a network drive on your Windows workstation to the directory /QIBM/UserData/WebAS5/Base/<InstanceName> on your iSeries.
2. Use the search tool from Windows Explorer to search for occurrence of your old node name (`ROCHAS07_diana` in our case) to the new node name. See Figure 10-9 for an example.

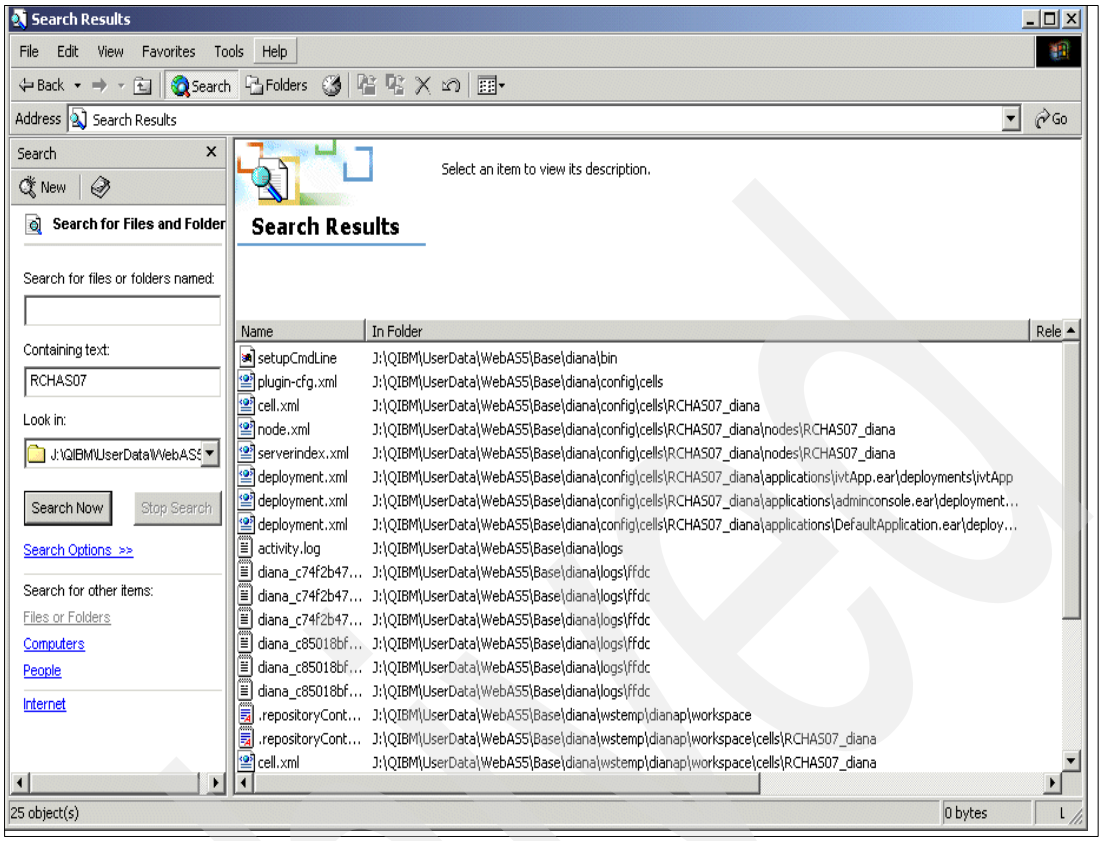


Figure 10-9 Example of the search results

3. Change the Node Name in the found files.

Attention: In the directory, /QIBM/UserData/WebAS5/Base/<InstanceName>/wstemp, you don't have to change any files. These files are created for each user profile. When you sign on again to the administrative console, they should be changed by WebSphere Application Server.

Starting WebSphere Application Server on the new system

Now you can start the WebSphere Application Server instance on the new system. For detailed information about this topic, see 6.5.3, "Starting a specific application server" on page 117.

10.6 PTF maintenance

WebSphere Application Server for iSeries requires periodic maintenance by applying the group PTFs for WebSphere. We show you how to verify that your system has the current level for each product related to WebSphere Application Server.

10.6.1 Cumulative PTFs

A cumulative PTF is a packaging of PTFs for all iSeries products. A new cumulative package is available about every 3-4 months. To determine the latest Cumulative PTFs available for the version of OS/400 installed on your iSeries, see the WebSphere Application Server Web site at:

http://www.ibm.com/eserver/iseries/software/websphere/wsappserver/services/service.htm

Note: The cumulative package number is a date. The first two digits are the year 01 = 2001. The next three digits are the cumulative available day in Julian format; for example, 100 = April 10.

To determine the current OS/400 cumulative PTF package installed on your server, perform the following steps:

1. Sign on to your iSeries.
2. Enter the Display PTF Status (DSPPTF) command on the OS/400 command line. The command displays output similar to that shown in Figure 10-10.

```
                          Display PTF Status
                                                          System:   MYSYSTEM
Product ID  . . . . . . . . . . . . . :   5722999
IPL source  . . . . . . . . . . . . . :   ##MACH#B
Release of base option  . . . . . . . :   V5R1M0 L00


Type options, press Enter.
  5=Display PTF details   6=Print cover letter   8=Display cover letter

     PTF                                                    IPL
Opt  ID       Status                                        Action
     TL02134  Temporarily applied                           None
     TL02071  Superseded                                    None
     TL02036  Superseded                                    None
     TL01302  Superseded                                    None
     TL01254  Superseded                                    None
     TL01226  Superseded                                    None
     TL01163  Superseded                                    None
     TL01114  Superseded                                    None
     TL01086  Superseded                                    None
                                                                      More...
F3=Exit   F11=Display alternate view   F17=Position to   F12=Cancel
```

Figure 10-10 Display PTF status

The first PTF listed with a status of `Temporarily applied` correlates to the cumulative PTF that is installed on the server. That is, PTF ID TL02134 indicates that cumulative (CUM) PTF package C02134510 is installed on this V5R1 server.

You must order and install the OS/400 cumulative PTF package periodically for preventive purposes.

10.6.2 Group PTFs

PTFs are also distributed as Group PTFs for iSeries. These group PTFs include all PTFs related to a particular iSeries product. There is a Group PTF for the WebSphere Application Server product. This group PTF includes the latest WebSphere Application Server PTFs that bring the product up to the latest WebSphere Application Server for iSeries level, as well as:

- IBM DB2 Universal Database
- IBM Developer Kit for Java
- IBM HTTP Server group PTFs

This group PTF also contains miscellaneous PTFs for IBM Developer Kit for Java, DB2 Universal Database for iSeries, WebSphere MQ, the WebSphere MQ classes for Java and JMS, and IBM HTTP Server, which are not included in other group PTFs or cumulative PTF packages, but must be installed.

You can find information about the latest group PTF for WAS at:

```
http://www-1.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/services/service.htm
```

10.6.3 Install PTFs

As part of your maintenance plan, you must apply PTFs periodically. Errors found in licensed programs are corrected with program temporary fixes (PTFs) that are released either as cumulative packages (CUM PTFs), group PTFs, or on an individual basis. Performance fixes for Java are released frequently, and you should ensure that you have them applied on your system.

The best way to do so is to apply the latest CUM PTF package for your version of OS/400. You should also apply the group PTFs for WebSphere Application Server, which include the group PTFs for Database, Java and HTTP server. The group PTFs frequently contain PTFs, which are not yet released in the cumulative PTF package.

Note: Group PTF for WAS may update the level of WebSphere Application Server on your system.

Use the Work with PTF Groups(WRKPTFGRP) command in V5R2 to determine the group PTF level on your system. For example, to display the group PTF for WebSphere Application Server V5R2, enter:

```
WRKPTFGRP PTFGRP(SF99245)
```

An example of the output created with the WRKPTFGRP command is shown in Figure 10-11.

```
                                Work with PTF Groups
                                                              System:   RCHAS02B
Type options, press Enter.
  4=Delete   5=Display   6=Print   9=Display related PTF groups

Opt  PTF Group             Level  Status
     SF99245                   1  Installed









                                                                        Bottom
F3=Exit   F6=Print   F11=Display descriptions   F12=Cancel
F22=Display entire field
```

Figure 10-11 An example of WRKPTFGRP command output

The number following the PTF Group identifier states the PTF level.

Note: This information is only available if a group PTF is installed.

The PTF groups to be verified are shown in Table 10-1 and Table 10-2.

Table 10-1 PTF Group for various program products

Program product	OS/400 V5R1	OS/400 V5R2
Java	SF99069	SF99169
HTTP Server	SF99156	SF99098
Database	SF99501	SF99502

The PTFs groups for WebSphere Application Server are described in Table 10-2.

Table 10-2 PTFs Group for WebSphere versions

WebSphere Application Server version	OS/400 V5R1	OS/400 V5R2
WebSphere Application Server 5.0	SF99243	SF99245
WebSphere Application Server Network Deployment 5.0	SF99244	SF99246

All product prerequisites must be installed before you install the group PTF package. If all prerequisites are not installed, WebSphere Application Server may fail when it is started.

Note: The WebSphere Application Server Network Deployment PTFs group includes the WebSphere Application Server Base PTFs group, so if you order the Network Deployment PTFs group, you do not need to order the Base group.

The following instructions describe how to install the WebSphere Application Server for the iSeries group PTF:

1. Verify that all of the prerequisite software is installed.
2. Place the WebSphere for iSeries group PTF CD-ROM into the CD-ROM drive on your iSeries server.
3. Sign on to your server. Your user profile must have *ALLOBJ authority.
4. Enter the following command to bring your system into a restricted state:

   ```
   ENDSBS SBS(*ALL)
   ```

5. Enter the following command from the OS/400 command line once the system is in a restricted state:

   ```
   GO PTF
   ```

6. Select option **8** (Install program temporary fix package) from the menu.
7. Specify the following parameter values and press Enter: (Figure 10-12 on page 430)
   a. Specify the device for your CD ROM drive (for example, OPT01)
   b. Automatic IPL: **N**

      Note: By specifying Automatic IPL: N, you are able to IPL the system when you are ready and have done all your backups. Do not attempt to use WebSphere Application Server until you have IPLed.

   c. PTF type: **1** (All PTFs)

```
                Install Options for Program Temporary Fixes
                                                        System: SYSTEMNAME
Type choices, press Enter.

  Device  . . . . . .   OPT01        Name, *SERVICE

  Automatic IPL . . . . . .   N      Y=Yes
                                     N=No

  Restart type  . . . . . .   *SYS   *SYS, *FULL

  PTF type  . . . . . .   1          1=All PTFs
                                     2=HIPER PTFs and HIPER LIC fixes
                                       only
                                     3=HIPER LIC fixes only
                                     4=Refresh Licensed Internal Code

  Other options . . . . . .   N      Y=Yes
                                     N=No



F3=Exit   F12=Cancel
```

Figure 10-12 Load PTFs

For information on the release as well as a description of known problems and workarounds, see the product Release Notes for the version of WebSphere that you are installing after installing the group PTF:

- Base edition:

  http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/docs/relnotes50.html

- Network Deployment edition:

  http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/docs/relnotes50nd.html

11

Troubleshooting

One of the most difficult tasks that you face when developing and implementing an application is what to do when it doesn't work correctly. In the case of developing WebSphere applications, a number of things can go wrong. Configuration errors can cause your application not to run at all, while programming errors can cause your application to run incorrectly. In this chapter we show you some of the key troubleshooting techniques that you can use to help solve such problems.

WebSphere Application Server offers several methods you can use to troubleshoot problems. Which method you use depends on the nature of the problem. Generally, you use a combination of these methods to determine the cause of a problem and then decide on an appropriate method for its resolution. Whether you are a beginner or experienced user, the following problem determination section leads you to resources and techniques to help you identify and respond to problems.

11.1 Where to look

When trying to solve a problem, there are a number of places to look for information:

- To find information about what happened, you can use the generated job logs, spooled files, or logs.
- To cause information to be generated about a running process or job, you can start a trace, run the job, and then examine the trace output.
- To follow the logic of program code, you can use the debug facility.

The first source of information for configuration and administration problems are the job logs, spooled files, and logs. If you cannot solve the problems using these files, try using a trace.

For runtime code problems, again look at the job logs, spooled files, and logs first. Next, use interactive debug. You have two options for debugging. The first choice is to use your Integrated Development Environment (like WebSphere Studio Application Developer). If this fails to isolate the problem, try debugging at the server. Finally, try running a trace.

11.2 Navigating through the problem

In order to navigate through a problem, you must first identify the area in which the problem is occurring and, from there, use the appropriate resources to determine what is causing the problem.

Generally, problems occurring within WebSphere Application Server for iSeries fall within one of the categories shown below.

11.2.1 Installation

If a problem occurs when you attempt to install the WebSphere Application Server for iSeries product, the first thing you should do is consult the installation and initial configuration documentation.

See the list below for the resources available for determining what the installation problem may be.

1. The WebSphere Application Server installation code outputs error messages when something goes wrong with the install. When you install locally from Qshell or use the RUNJVA command, the output is displayed to the screen. When you install remotely from a workstation, the error messages are displayed to the screen from which you launched the install.

   If an OS/400 message is associated with the error, the message identifier is displayed in the error message. To get more information about the message, use the Display Message Description (DSPMSGD) command from an OS/400 command line.

2. Check the display log for message related to the installation. You can use the LICPGM menu.

   a. Type `go licpgm` from a command line

   b. Select option **50**.

   c. The system asks for the start date and time to look into log files. Enter the appropriate values.

d. The Display History Log Contents window appears with the messages related to the installation. You can use the F1 key in order to have more information about each message. The message identifier could help you to find out the problem.

3. Check the WebSphere Application Server 5.0 for iSeries Frequently Asked Questions (FAQ) database at:

   http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/docs/faq.htm

4. Refer to the WebSphere Application Server for iSeries newsgroup. This iSeries Technical Support Web-based forum is dedicated to WebSphere Application Server for iSeries:

   http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/services/forum.htm

5. Check the Software Knowledge Base at:

   http://www-912.ibm.com/s_dir/slkbase.nsf/slkbase

6. Contact IBM support.

Tips:

- The user profile used by the installation process must have *ALLOBJ and *SECADM special authorization. Otherwise, the installation of WebSphere Application Server will fail.
- You must have already installed JDK 1.3 on your iSeries system.
- You should check your OS/400 installed version. WebSphere Application Server can be installed only on OS/400 V5R1 and V5R2.

11.2.2 WebSphere Application Server startup

If a problem occurs when you try to start the WebSphere Application Server for iSeries product, the first thing you should do is consult the installation and initial configuration documentation.

The resources listed below can help you determine what the startup problem may be.

1. In 6.3, “Quick start tutorial” on page 110, we step you through the system configuration and startup process.
2. Check the job log of the failing WebSphere Application Server job:

   If the application server job (Server1 by default) fails to start, examine the job log for errors. For information on viewing job logs, see 11.3, “Resources for identifying problems” on page 444.

3. Check the WebSphere Application Server log files for errors:

   If the application server job fails to start, check the JVM *system output* and *system error* logs, IBM services logs on activity log and process logs on standard error and standard output logs. For information on the log files and where they are located, see 11.4, “WebSphere Application Server log files” on page 461.

11.2.3 Administrative Console connection problems

If the administrative console fails to start, see 6.7, “Working with the Administrative Console” on page 159. If you cannot resolve the problem, follow this troubleshooting to determine the cause of the problem when attempting to use the administrative console or the WebSphere admin scripting tools (wsadmin).

1. Administrative functions are supported by the application server (server1 by default) for a WebSphere Application Server instance, or the Deployment Manager (such as “dmgr”) for the default Network Deployment instance.

   The server process must be running in order to use the administrative console. For wsadmin, many functions require a connection to the server as well. Verify that the server is running and ready to receive administrative requests. For more information on how to verify that the server is running, see 6.5.4, “Verifying that the WAS environment has started” on page 119.

   If the server is not started and you are using wsadmin for an administrative function that does not require the server to be started, ensure that you have specified `-conntype NONE` when invoking wsadmin.

2. Use the TCP/IP ping command to test that the hostname where the application server or Deployment Manager is running is reachable from the system where the browser or wsadmin program are being used. You must check your firewall configuration in order to assure that the port selected for the administrative console is available to use. For a detailed troubleshooting of this topic, see “Verifying TCP/IP configuration for Administrative Console” on page 435.

3. If the host where the application server or Deployment Manager is running is remote to the machine from which the client browser or wsadmin command is running, ensure that the hostname in the browser URL for the console is correct, or the `-host hostname` option of the wsadmin command is being used to direct wsadmin to the right server.

4. If you are using the wsadmin tool, see 9.2, “Configuring and launching wsadmin” on page 386. Also check the wsadmin log files located in the logs directory of your instance (by default) for errors.

5. Check the log files for the server for errors. For information on the log files and where they are located, see 11.3, “Resources for identifying problems” on page 444.

6. Read the WebSphere Application Server Release Notes at these sites:
   - Base edition:

     `http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/docs/relnotes50.html`

   - Network Deployment edition:

     `http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/docs/relnotes50nd.html`

7. Check the WebSphere Server FAQ(Frequently Asked Question) at:

   `http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/docs/faq.htm`

8. Refer to the WebSphere Application Server for iSeries newsgroup at:

   `http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/services/forum.htm`

9. Check the Software Knowledge Base at:

   `http://www-912.ibm.com/s_dir/slkbase.nsf/slkbase`

Verifying TCP/IP configuration for Administrative Console

In this section we guide you through the process of verifying the TCP/IP connection between your iSeries and the workstation selected as a console. Table 11-1 shows the example values that we use on the following steps.

Table 11-1 Example values

iSeries Host Name	MyiSeries.ibm.com (note the mixed case)
iSeries IP Address	9.11.22.33
iSeries Host Name	MyiSeries.ibm.com
Administrative Console Port	9090
PC Host Name	MyPC.ibm.com
PC IP Address	9.1.2.3

1. Verify the TCP/IP configuration information:
   a. Enter the **CFGTCP** (Configure TCP/IP) command on the iSeries server.
   b. Select option **12**.

      You should see these values:

      - Host name: 'MyiSeries' (quotes are included)
      - Domain name: 'ibm.com' (quotes are included)
      - The values for the Hostname search priority and Domain name server are important when figuring out what is causing a problem. If search priority is *LOCAL, the host table is being used to resolve IP addresses. If the value is *REMOTE, a DNS is used first, and the host table is then used if the DNS cannot resolve the name. Figure 11-1 shows an example of what you should see.

```
                    Change TCP/IP Domain (CHGTCPDMN)

Type choices, press Enter.

Host name  . . . . . .  'MyiSeries'

Domain name  . . . . . .  'IBM.COM'



Domain search list . . . . . .  'IBM.COM COMPANY.IBM.COM'



Host name search priority  . . .  *LOCAL        *REMOTE, *LOCAL, *SAME
Domain name server:
  Internet address . . . . . .  '9.11.10.10'
                                ''
                                ''

                                                                    Bottom
F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
F13=How to use this display        F24=More keys
```

Figure 11-1 Example of TCP/IP Domain on iSeries

c. Press F12.

d. Select option **10.**

If the address is in the list, it should look as shown in Figure 11-2:

```
                    Work with TCP/IP Host Table Entries
                                                                System:   AS07
Type options, press Enter.
  1=Add   2=Change   4=Remove   5=Display   7=Rename

     Internet         Host
Opt  Address          Name

     9.11.22.33       MYISERIES.IBM.COM
                      MYSERIES
     127.0.0.1        LOOPBACK
                      LOCALHOST

                                                                        Bottom
F3=Exit   F5=Refresh   F6=Print list   F12=Cancel   F17=Position to
```

Figure 11-2 Example of Host table entries

You must have at least one host name (either the unqualified or the qualified) in this list, or you must have a DNS to resolve the host name if you don't have it defined in your iSeries system.

e. Press F12.

f. Select option **1**.

You should see IP `9.11.22.33` in the list. Press F11, and you should see that it is active.

2. Run the IPTest utility on the iSeries server.

a. Start the QShell environment. Enter: **`qsh`**

b. Execute the following command:

```
cd /qibm/proddata/webas5/base/bin
```

c. Enter: `java IPTest`
You should get results like those shown in Table 11-3.

```
                                QSH Command Entry
 $
> cd qibm
 $
> cd proddata
 $
> cd webas5
 $
> cd base
 $
> cd bin
 $
> java IPTest
 Local Address: 9.11.22.33
 Local Name: MYISERIES.IBM.COM
 All addresses for MYISERIES.IBM.COM:
      9.11.22.33
 $

===>



F3=Exit  F6=Print F9=Retrieve F12=Disconnect
 F13=Clear F17=Top  F18=Bottom  F21=CL command entry
```

Figure 11-3 Example java IPTest

3. Ping the client from the iSeries server:

   a. `ping MyPC`
   You should output like this:

```
Verifying connection to host system MyPC.ibm.com at address 9.1.2.3.
PING reply 1 from 9.1.2.3 took 46 ms. 256 bytes. TTL 125.
PING reply 2 from 9.1.2.3 took 6 ms. 256 bytes. TTL 125.
PING reply 3 from 9.1.2.3 took 7 ms. 256 bytes. TTL 125.
PING reply 4 from 9.1.2.3 took 11 ms. 256 bytes. TTL 125.
PING reply 5 from 9.1.2.3 took 11 ms. 256 bytes. TTL 125.
Round-trip (in milliseconds) min/avg/max = 6/16/46
Connection verification statistics: 5 of 5 successful (100%).
```

4. Ping the iSeries server from the iSeries server:

   a. On the iSeries server, enter: `ping MyiSeries`
   You should see output like this:

```
Verifying connection to host system MyiSeries.ibm.com at address 9.11.22.33.
PING reply 1 from 9.11.22.33 took 0 ms. 256 bytes. TTL 64.
PING reply 2 from 9.11.22.33 took 0 ms. 256 bytes. TTL 64.
PING reply 3 from 9.11.22.33 took 0 ms. 256 bytes. TTL 64.
PING reply 4 from 9.11.22.33 took 0 ms. 256 bytes. TTL 64.
PING reply 5 from 9.11.22.33 took 0 ms. 256 bytes. TTL 64.
Round-trip (in milliseconds) min/avg/max = 0/0/0
Connection verification statistics: 5 of 5 successful (100%).
```

5. Verify that the WebSphere Application Server is running on the iSeries server.

   a. **`WRKACTJOB SBS(QEJBAS5)`**
   You should see your Application server job. The default name is SERVER1.

b. Enter option **5** next to the administrative server job, then enter option **10.**
You should see a message in the job log like this:

```
WebSphere application server serverName ready.
```

c. Press F1 on this message. You should see a message like that shown in Figure 11-4.

```
                        Additional Message Information

Message ID . . . . . :   EJB0106       Severity . . . . . :   00
Message type . . . . . :   Information
Date sent  . . . . . :   11/14/02      Time sent  . . . . . :   15:44:46

Message . . . . :   WebSphere application server server1 ready.
Cause . . . . . :   WebSphere application server server1 in job
  025941/QEJBSVR/SERVER1 is ready to handle administrative requests on port
  9090.

                                                                       Bottom
Press Enter to continue.

F3=Exit   F6=Print   F9=Display message details   F12=Cancel
F21=Select assistance level
```

Figure 11-4 dspjoblog of SERVER1 job

6. Verify that the server is listening on the right ports on the iSeries server:
   a. Enter the Network Status command (NETSTAT) and select option **3**.
   b. Use option **5** to display the connection where either the local or remote port is 9090. Verify that the local host name and the remote host name are either “MyPC.ibm.com” or “MyiSeries.ibm.com”.
7. Ping the iSeries server from the workstation.
   a. From a workstation command line, enter: `ping MyiSeries`
   You should see output like this:

```
Pinging MyiSeries.ibm.com [9.11.22.33] with 32 bytes of data:
Reply from 9.11.22.33: bytes=32 time=10ms TTL=61
Reply from 9.11.22.33: bytes=32 time=10ms TTL=61
Reply from 9.11.22.33: bytes=32 time<10ms TTL=61
Reply from 9.11.22.33: bytes=32 time=10ms TTL=61
```

8. Verify the hosts file on the workstation.
   a. On Windows NT or Windows 2000, the host file is the <Drive>:\WINNT\system32\drivers\etc\hosts file where <Drive> is the drive letter where Windows NT is installed.
   If the hosts file contains an entry for the iSeries server, it should look like this:

```
9.11.22.33      MyiSeries.ibm.com
9.11.22.33      MyiSeries
```

   It is not necessary to have a host file entry if the iSeries server can be correctly resolved by a Domain Name Server (DNS).

9. **Other tips:** The workstation client should be able to connect with the unqualified host name, host name qualified by domain name, or IP address, assuming the client can ping all these, with these caveats:
   - If the host is in a different domain, the unqualified host name will not work unless there is a matching entry in the workstation hosts file.
   - The IP address only works if it resolves to a host name which matches the correct case of the server name. If not (for example, ping IP Address does not show the server name with correct case), then adding an entry to the hosts file with the correct server name case should correct the problem.

11.2.4 HTTP server startup

If a problem occurs when you try to start the HTTP server instance in your system, the first thing you should do is consult the installation and initial configuration documentation.

The resources listed below are available to help you determine what the startup problem may be.

HTTP server instance job logs

You can check the job log for your HTTP instances. Basically, there are many jobs with the same name of your HTTP instances in the QHTTPSVR subsystem. Each job has a different purpose: one of them is the manager job for HTTP instance, there are two jobs for logging purposes (if the error logging option is available for the HTTP server instance) and one additional job is the primary job for HTTP server instance. If you want to see more information about the HTTP Server (powered by Apache), refer to the IBM Redbook *HTTP Server (Powered by Apache): An Integrated Solution for IBM iSeries Servers*, SG24-6716 at:

```
http://www.redbooks.ibm.com/pubs/pdfs/redbooks/sg246716.pdf
```

If you are working with Domino HTTP server, you can review its joblog for the job HTTP within a subsystem name DOMINOxx.

HTTP server for iSeries (Powered by Apache) trace log

The IBM HTTP Server for iSeries (powered by Apache) generates a trace file. You must start the HTTP server in a mode that outputs trace information. There are various levels of trace information related to a specific server:

1. -ve (error) for a trace that contains records for all error return codes or exception conditions
2. -vi (information) for a trace that contains -ve level trace records as well as trace records for entry and exit points from application level API's and API parameters
3. -vv (verbose) for a trace that contains -vi level trace records as well as trace records for debugging control flow or data corruption

For example, if you are using option -vv:

```
STRTCPSVR SERVER(*HTTP) HTTPSVR(<serverInstanceName> '-vv')
```

Here, `<serverInstanceName>` is the name of your HTTP server instance.

There are three ways to get output from the trace:

1. **ENDTCPSVR command:** When the server has ended, the trace data is placed into a spool file. There is a spool file for each job that is running on the server. If a server ends abnormally, trace data is placed into spool files even if tracing is not active at the time of the error. The spool file name is QZSRHTTPPTR, and there is a spool file for each job associated with your HTTP server instance. Figure 11-5 shows an example of the HTTP server instance trace log.
2. **DMPUSRTRC command:** This command dumps the trace data for a specific job to the display or to a physical file member in the QTEMP library. For example:
   a. Use the WRKACTJOB command to find the server job number. For example:
      `WRKACTJOB SBS(QHTTPSVR)`
   b. Dump the user trace to a file in QTEMP. For example:
      `DMPUSRTRC JOB(nnnnnn/QTMHHTTP/WEBSERVER2)`
      Here, nnnnnn is the job number and `WEBSERVER2` is the HTTP server instance.
   c. Use the DSPPFM command to view the contents of the trace. For example:

      ```
      DSPPFM QTEMP/QAPOZDMP MBR(QPOZnnnnnn)
      ```
3. **Trctcpapp command:** You can use the TRCTCPAPP command to initiate a trace after the server is started and to end a trace. To use the TRCTCPAPP command, the server must have been started with the STRTCPSVR command. If you start your server using the SBMJOB or SPAWN commands, you cannot use TRCTCPAPP to start or stop traces. TRCTCPAPP SET (*OFF) produces a spool file for each job that is running on the server. The spool file name is QZSRHTTPPTR, and there is a spool file for each job associated with your HTTP server instance. Figure 11-5 shows an example of the HTTP server instance trace log.
4. If you started the trace with the STRTCPSVR and one of the trace startup parameters (-ve, -vi, or -vv), then you must do the following to end the trace:
   a. Enter the TRCTCPAPP SET (*ON) command to synchronize it with the STRTCPSVR command. For example:
      `TRCTCPAPP APP(HTTP) SET(*ON) HTTPSVR(WASSERVER) TRCLVL(*VERBOSE)`
   b. Enter the TRCTCPAPP SET (*OFF) command. For example:
      `TRCTCPAPP APP(*HTTP) SET (*OFF) TITLE('My title')`

```
                                Display Spooled File
 File  . . . . . :  QZSRHTTPTR
Page/Line   1/6
 Control . . . . .
Columns     1 - 130
 Find  . . . . . .

*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+...
.9....+....0....+....1....+....2....+....3
User Trace Dump for job 029057/QTMHHTTP/WEBSERVER2. Size: 16382K, Wrapped 0 times.
 --- 11/25/2002 11:36:48 ---
    00000003:348544
 Application.............: HTTP
 Instance................: WEBSERVER2
 System..................: RCHAS07.ITSO.IBM.COM
 Start date/time.........: Mon Nov 25 11:36:48 2002
    00000003:348584
 THREAD ID:TIMESTAMP  DATA
 -------------------------------------------------
  00000003:348600 CHILD: Trace level set to 'VERBOSE' for this process.
  00000003:348616 Apache/2.0.39
  00000003:348640 Module Magic Number: 20020612
  00000003:348688 isBidiCcsid() - ccsid 37 is not a bidirectional ccsid
  00000003:348712 isBidiCcsid() - ccsid 37 is not a bidirectional ccsid
  00000003:348928 createCcsidPairSingleByte() - CCSID 1 is 37, CCSID 2 is 37
  00000003:348976 isBidiCcsid() - ccsid 37 is not a bidirectional ccsid
  00000003:348992 isBidiCcsid() - ccsid 37 is not a bidirectional ccsid
  00000003:349048 isBidiCcsid() - ccsid 37 is not a bidirectional ccsid

More...
 F3=Exit  F12=Cancel  F19=Left  F20=Right  F24=More keys
 Record requested before first record of file.
```

Figure 11-5 Example of HTTP server instance trace log

HTTP server instance error log

You can find additional log information by enabling the IBM HTTP server for iSeries (powered by Apache) error log. This log records client request error, such as timing out or access denial. To enable logging, perform these steps:

1. Make sure the HTTP Server ADMIN instance is running.
   a. In the WRKACTJOB display, look for the ADMIN job under the QHTTPSVR subsystem.
   b. If the job does not display, start the ADMIN server with the following command:

      ```
      STRTCPSVR SERVER(*HTTP) HTTPSVR(*ADMIN)
      ```

2. Enter the following URL in a JavaScript-enabled browser:

   ```
   http://your.host.name:2001
   ```

   Here, `your.host.name` is the domain name of your HTTP server.
3. Click **IBM HTTP Server for iSeries**.
4. Select the **Manage** tab.
5. Select your HTTP server instance's configuration name from the **Server** pull-down menu.
6. In the left-hand pane, ensure **Server Properties** is expanded.
7. Under **Server Properties**, click **Logging.** You may have to scroll down to find a link.

8. Select **Error logs** tab.
9. In the main window, set up the error log. (The help text explains the meaning of the values in the page. To view the help text, click the **?** icon.) Click **Apply** (see Figure 11-6).
10. Stop your HTTP server instance and restart it.

The HTTP server changes the name you specify for the log file. For example, if you name the file `myerror.log`, the HTTP server generates the file with the name `myerror.log.Q`*yyymmdd*, where *yyy* is the last three digits of the year, *mm* is the month, and *dd* is the day of the month (for example, `myerror.log.Q0991005`). To view the log file, you can use the iSeries Edit File Utility (EDTF). The file is located in the HTTP instance's IFS directory subtree.

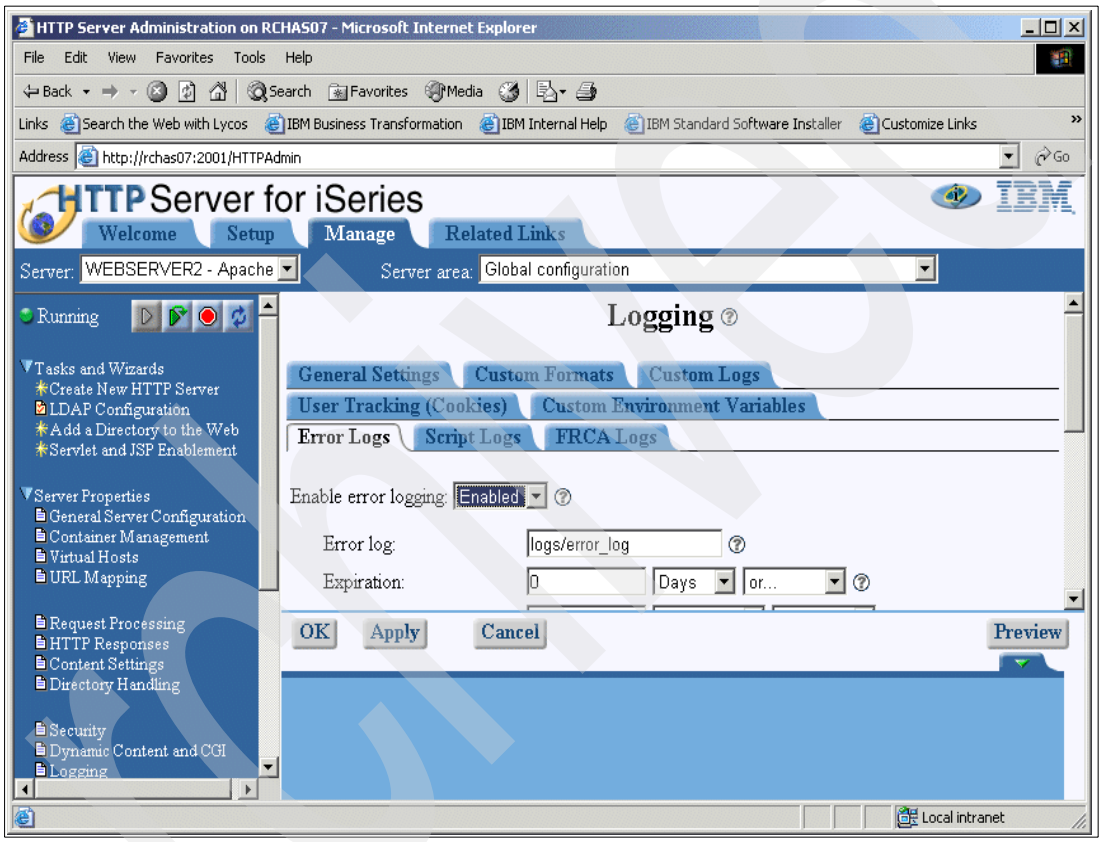


Figure 11-6 Example of the HTTP server instance logging

Web server plug-in log

The plug-in log records HTTP server-side processing and servlet request routing between the HTTP server and WebSphere Application Server (the plug-in connects the HTTP server and WebSphere application Server together). You can manage this log by editing the application server's plugin-cfg.xml files, which is located in the following directory, where `<InstanceName>` is the name of your instance:

`/QIBM/UserData/WebAS5/Base/<InstanceName>/config/cells`

The plugin-cfg.xml file contains a tag near the beginning named Log. Log has two attributes:

- *LogLevel* specifies amount and type of information logged to a file. Valid values are:
  - Trace
  - Warn
  - Error
- *Name* specifies the location and name of the file where logging information is written.

You can see an example of the plugin-cfg.xml file in Figure 11-7.

Attention: QTMHHTTP user must have these authorizations, otherwise the startup for your HTTP server instance will fail:

Execute authorization for:

```
/QIBM/UserData/WebAS5/Base/<InstanceName>
/QIBM/UserData/WebAS5/Base/<InstanceName>/config
/QIBM/UserData/WebAS5/Base/<InstanceName>myserver /config/cells
```

Read and execute authorization for the plug-in file:

```
/QIBM/UserData/WebAS5/Base/<InstanceName>/config/cells/plugin-cfg.xml
```

Here, `<InstanceName>` is the name for your WebSphere Application Server instance.

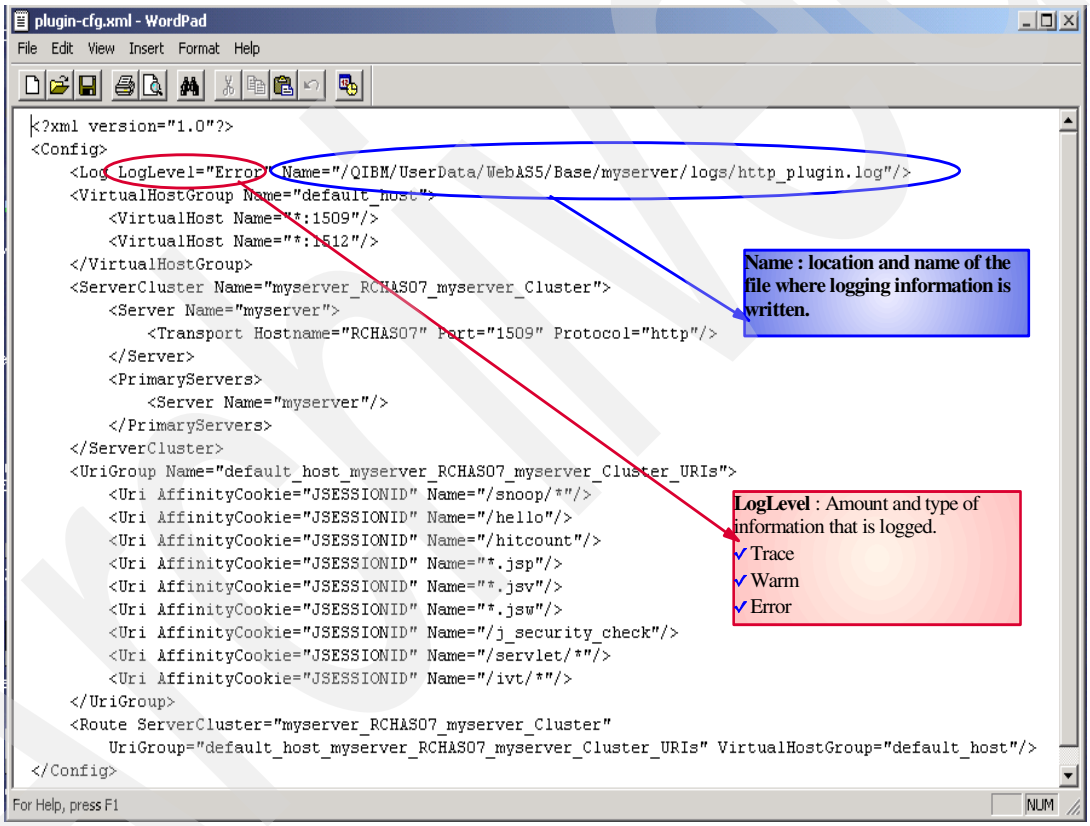


Figure 11-7 Example of plugin-cfg.xml

WebSphere Application Server log files

For information about WebSphere Application Server log files, see 11.4, “WebSphere Application Server log files” on page 461.

11.2.5 WebSphere Application Server configuration

A number of resources are available for determining what may be causing a problem when you are attempting to install or configure resources from the WebSphere administrative console.

The following resources are available for determining what the problem might be:

1. If a problem occurs when you are installing or configuring a resource, typically an error page or message containing error information is displayed. Click the **Details** link, if there is one, to view more information on the possible cause of the problem.
2. For information on the error, check the application server standard output and standard error log files. For information on the log files and where they are located, see 11.3, “Resources for identifying problems” on page 444.
3. Read the WebSphere Application Server Release Notes at one of these sites:
   - Base edition:

     http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/docs/relnotes50.html
   - Network Deployment edition:

     http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/docs/relnotes50nd.html
4. Check the WebSphere Server FAQ(Frequently Asked Question) at:

   http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/docs/faq.htm
5. Refer to the WebSphere Application Server for iSeries newsgroup at:

   http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/services/forum.htm

11.3 Resources for identifying problems

A number of jobs make up the WebSphere Application Server runtime. You can use iSeries commands to monitor them. Message queue support is new for WebSphere Application Server. This support allows you to specify a message queue to which messages that are sent to the administrative server and application server job logs are also sent.

The WebSphere Application Server for iSeries product is shipped configured to use a single WebSphere instance, the default instance. However, multiple instances are also supported. See 6.5.2, “Multiple instances of WebSphere Application Server” on page 113 for more information. The following sections describe the jobs used by a WebSphere instance, how you can monitor the jobs using a message queue, and how you can monitor the jobs by using iSeries command language (CL) commands.

11.3.1 WebSphere Application Server iSeries jobs

Each WebSphere Application Server instance consists of one or more jobs. For the WebSphere Application Server product, these jobs run in the QEJBAS5 subsystem.

- **application server job:** The job name is the first 10 characters of the application server name. If the instance is not part of a Network Deployment domain, and the instance is not using embedded JMS, this is the only job started for an instance. For more information see 2.17.2, “WebSphere Application Server job” on page 44.
- **MQ listener job:** If the instance is configured to use the embedded JMS server, this job is used by the embedded JMS provider to establish connections to the embedded JMS server. The job name is `QEJBMQLSR`. If you are using multiple instances of WebSphere Application Server with embedded JMS enabled, you see a `QEJBMQLSR` job for each instance. For more information, see 2.17.3, “MQ listener job” on page 45.

- **Node agent job:** The node agent job exists only if the instance is managed by a Network Deployment Manager (in other words, it is part of a Network Deployment domain). The job name for the node agent process is `NODEAGENT`. If you are using multiple instances of WebSphere Application Server and the instances have been added to a Network Deployment cell, you see one `NODAGENT` job for each instance. You can determine which `NODEAGENT` job is for your instance as described in "Verifying that the WebSphere Application Server node agent has started" on page 292.

Each WebSphere Application Server Network Deployment instance consists of one job for the deployment manager. This job runs in the QEJBASND5 subsystem:

- **Deployment Manager server job:** In a Network Deployment instance, there is a single job for the deployment manager. The job name is the first 10 characters of the application server name for the Deployment Manager. If the first 10 characters do not provide a valid iSeries job name, the WebSphere Application Server runtime creates a valid job name. If the runtime cannot create a valid job name for the application server, it uses the default job name QEJBSVR. The Network Deployment product is shipped and configured to use the default instance. The job name for the default Deployment Manager instance is *DMGR*. For more information, also see "Verifying that the Deployment Manager has started" on page 275.

11.3.2 Other jobs

In addition to the jobs that run in the QEJBAS5 or QEJBASND5 subsystems, WebSphere Application Server instances use other jobs.

Jobs for embedded JMS in subsystem QMQM

If the instance is configured to use the embedded JMS server, the following jobs are started in the QMQM subsystem when your application server or node agent server is started. When the application server job is ended, these jobs are ended also:

- AMQALMPX
- AMQPCSEA
- AMQRMPPA
- AMQRRMFA
- AMQZDMAA
- AMQZLAA0
- AMQZXMA0
- RUNMQCHI

For more information, see "Other jobs for embedded JMS" on page 45.

Jobs depending database access from your application running in WAS

1. If you are using the IBM Developer Kit for Java JDBC driver for database access, your instance uses one or more QSQSRVR jobs that run in the QSYSWRK subsystem.

   To determine what QSQSRVR job your application server job is using, view the joblog for the server job you are interested in. For each JDBC connection obtained, you will see message SQL7908 with message text similar to this: **"**`Job 163707/QUSER/QSQSRVR used for SQL server mode processing`**."**

   The QSQSRVR jobs are prestarted jobs that are created and reused as necessary. The QSQSRVR jobs being used by an application are displayed in the job log for that application. The job information for each QSQSRVR job is written to the job log of the application server job.

The QSQSRVR jobs may contain valuable information if a database error occurs. Once an application job ends, any QSQSRVR jobs associated with the application job are recycled. The job log is written out or cleared. It is easiest to find the QSQSRVR job log while it is still associated with the application server job. To view the job log, use the Display job log (DSPJOBLOG) command:

```
DSPJOBLOG JOB(qsqsrvr_job_number/QUSER/QSQSRVR) OUTPUT(*PRINT)
```

Once a job has ended, it is more difficult to determine which is the correct job log to view. There will be multiple job logs for a single QSQSRVR job. Make note of the time the Job nnnn used for SQL server processing message was sent to the job you are debugging. Use the Work Spooled Files (WRKSPLF) command to find the job logs:

```
WRKSPLF SELECT(profile_name *ALL *ALL QSQSRVR)
```

In this case, *profile_name* is the name of the user profile under which the job using the QSQSRVR jobs was running (QEJBSVR is the default user profile for application server). A list of job logs is displayed. Display each job log until you find the job log for the QSQSRVR job you want, which has a start time that matches the time the SQL server processing message was sent. There are several normal messages that are in a QSQSRVR job log associated with WebSphere Application Server for iSeries.Here is an example of displaying the QSQSRVR job log. The command used is:

```
DSPJOBLOG JOB(101812/Quser/QSQSRVR) OUTPUT(*)
```

The output is shown in Figure 11-8.

```
 Job 258489/QUSER/QSQSRVR started on 03/29/01 at 14:03:48 i
 QSYSWRK in QSYS. Job entered system on 03/29/01 at 14:03
User Profile = QEJB
Commit level *RR escalated to *EXCL lock.
Commit level *RR escalated to *EXCL lock.
Commit level *RR escalated to *EXCL lock.
```

Figure 11-8 QSQSRVR job log

2. If you are using the IBM Toolbox for Java JDBC driver for database access, your instance uses one or more QZDASOINIT jobs that run in subsystem QUSRWRK.

   The QZDASOINIT jobs are prestarted jobs that are created and reused as necessary. The QZDASOINIT jobs may contain valuable information if a database error occurs. Once an application job ends, any QZDASOINIT jobs associated with the application job are recycled. The job log is written out or cleared. To view the job log, use the Display job log (DSPJOBLOG) command:

   ```
   DSPJOBLOG JOB(qzdasoinit_job_number/QUSER/QZDASOINIT) OUTPUT(*PRINT)
   ```

   Once a job has ended, it is more difficult to determine which is the correct job log to view. There will be multiple job logs for a single QZDASOINIT job. Use the Work Spooled Files (WRKSPLF) command to find the job logs:

   ```
   WRKSPLF SELECT(QUSER *ALL *ALL QZDASOINIT)
   ```

   A list of job logs is displayed. Display each job log until you find the job log for the QZDASOINIT job you want, which has a start time that matches the time the SQL server processing message was sent.

Jobs for Web server

Depending on which Web server you are using to serve Web resources, your instance uses jobs from the Web server instance:

- **IBM HTTP Server (powered by Apache):** Each IBM HTTP Server (powered by Apache) instance consists of two or more jobs that run in the QHTTPSVR subsystem. The name of each job is the same as the name of your HTTP server instance. The WebSphere Application Server Web server plugin code runs in the second job listed (via WRKACTJOB SBS(QHTTPSVR)) for your HTTP server instance.
- **Domino Web Server:** Each Domino Web Server instance has a corresponding subsystem in which the jobs for the instance run. The subsystem for the first Web server instance created is DOMINO001, the subsystem for the second Web server instance created is DOMINO002, and so on. The WebSphere Application Server Web server plugin code runs in the job named HTTP in the subsystem in which your Web server instance is running.

11.3.3 Using a message queue to monitor WebSphere Application Server

WebSphere Application Server for iSeries allows you to specify an iSeries message queue object to which product messages are sent. The product messages are the same WebSphere Application Server messages that are sent to the joblog of the application server job.

To enable the message queue support for an application server, use the administrative console to specify the **os400.websphere.message.queue** system property for your server. The message queue is specified by using the Integrated File System pathname of the object. For message queues in library QSYS, the format is /QSYS.LIB/messageQueue.MSGQ. For message queues in libraries other than QSYS, the format is /QSYS.LIB/yourLib.LIB/messageQueue.MSGQ. The message queue must exist and the QEJBSVR user profile must have *CHANGE authority to the message queue.

Follow these steps to specify the os400.websphere.message.queue system property for your application server:

1. Start the administrative console.
2. Expand Servers and click **Application Servers**.
3. Click the link for the application server you wish to change.
4. Scroll down in the right-hand frame and click **Process Definition**.
5. Under the Additional Properties section, click **Java Virtual Machine**.
6. Scroll down and click **Custom Properties** under the Additional Properties section in the right-hand frame.
7. Click **New**.
8. Enter os400.websphere.message.queue in the Name field and enter the integrated file system pathname for the message queue in the Value field. For example, you would specify `/QSYS.LIB/QGPL.LIB/MYMSGQ.MSGQ` to use the `MYMSGQ` message queue in the QGPL library.
9. Click **OK**.
10. Click **Save** at the top of the right-hand frame to save the configuration change.
11. On the Save page, click **Save** to save the changes.
12. Stop and start your WebSphere Application Server instance.

Note: You can enable the message queue support for any WebSphere Application Server server such as a Network Deployment Manager, a node agent, or a JMS server. Simply use the administrative console to locate the server and then follow steps 3 through 11 above

Figure 11-9 shows an example of the administrative console.

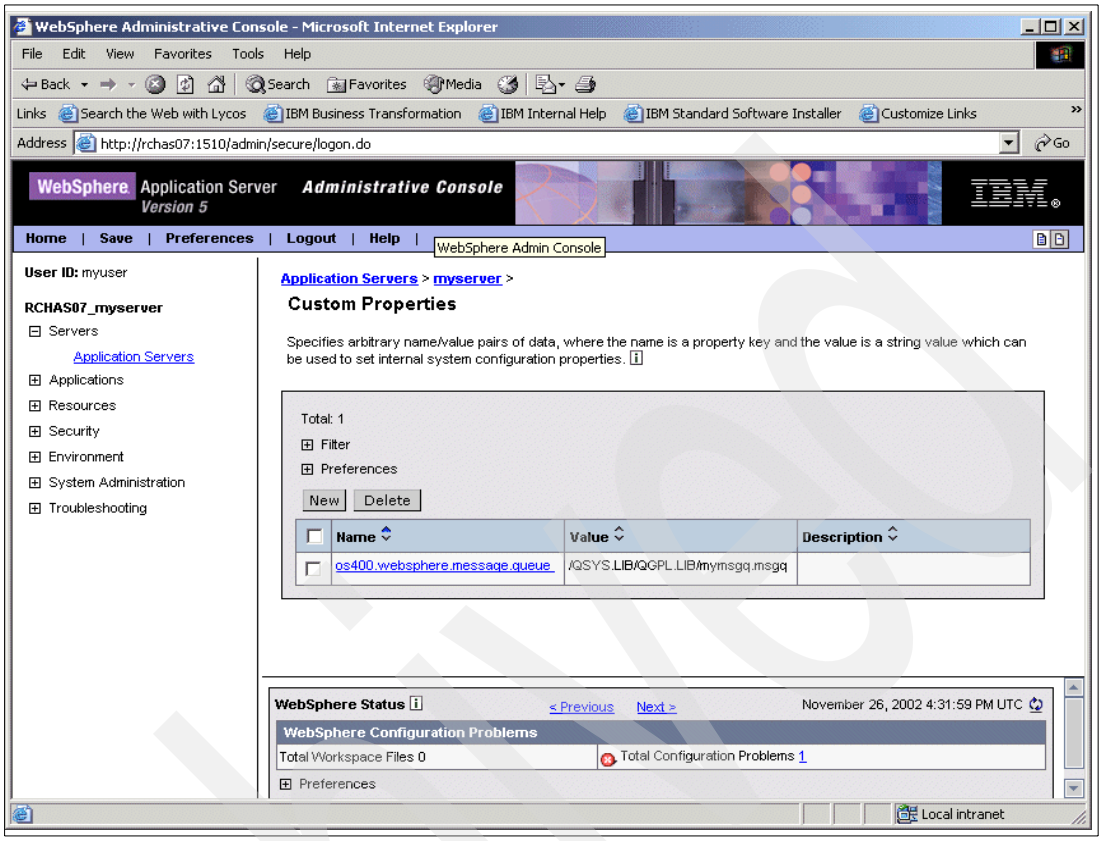


Figure 11-9 Example of message queue monitoring for WebSphere Application Server instance

After this definition is done, you can see the messages for your WebSphere Application Server instance, in the message queue defined. Figure 11-10 shows an example of how these message appear.

```
                          Display Messages
                                                        System:   AS07
Queue . . . . . :   MYMSGQ                  Program . . . . :   *DSPMSG
  Library . . . :     QGPL                    Library . . . :
Severity  . . . :   00                      Delivery  . . . :   *HOLD

Type reply (if required), press Enter.
  Server starting with user profile QEJBSVR and JDK 1.3.1.
  WebSphere application server myserver ready.




                                                                    Bottom
F3=Exit              F11=Remove a message                F12=Cancel
F13=Remove all       F16=Remove all except unanswered    F24=More keys
```

Figure 11-10 Example of DSPMSGQ

11.3.4 OS/400 command language (CL) commands for monitoring jobs

Use iSeries command language (CL) commands to monitor jobs that run in your WebSphere Application Server environment. You can view job logs, display message details, and view spooled files associated with the jobs. The job logs and associated spooled files for the jobs can contain valuable information for determining the root of a problem.

Note: For more information on iSeries work management CL commands, see OS/400 Work Management V5R1 (SC41-5306-03) or iSeries Work Management V5R2.

Table 11-2 lists some iSeries commands that you can use to monitor WebSphere Application Server jobs:

Table 11-2 CL commands

Task	iSeries command and description

Work with Active WebSphere Server jobs	For WebSphere Application Server V5.0 jobs that run in the QEJBAS5 subsystem:
	`WRKACTJOB SBS(QEJBAS5)`
	For WebSphere Application Server V5.0 jobs that run in the QEJBASND5 subsystem:
	`WRKACTJOB SBS(QEJBASND5)`

Work with all jobs with a specific name	`WRKJOB JOB(job_name)` This command lists all jobs, active or not, named job_name. The jobs are listed by date, most recent first. This command is very useful when a WebSphere Application Server job ends abnormally or fails to start successfully, and you wish to view the joblog. For instance, if the application server job is starting and then ending immediately, you could do this to view its joblog: 1. Run the command `WRKJOB JOB(SERVER1)`. 2. Specify option **1** on the option line next to the first job listed that has a status of OUTQ. 3. Specify option **4** (Work with spooled files) on the command line of the Work with Job screen. 4. Specify option **5** next to the QPJOBLOG spooled file to view the contents.
Work with a specific job	`WRKJOB JOB(job_number/job_user/job_name)` This command displays the Work with Job screen for the specified job. This command is very useful when you know the fully qualified job information for the job.

View the job log for an active WebSphere server job	`WRKACTJOB SBS(QEJBAS5)`
	or
	`WRKACTJOB SBS(QEJBASND5)`
	1. Run one of the commands above.
	2. Enter option **5** (Work with) in the option line next to the active job whose job log you want to view.
	3. In the Work with Job display, enter option **10** (Display job log, if active or on job queue).
	4. Press F10 to display all messages.
	5. In the job log, position the cursor on a message for which you want to display extended message information.
	6. Press F1 (Help).
	Similarly, you can view ended jobs by using the Work with User Jobs (WRKUSRJOB) or Work with Job (WRKJOB) command.

Task	iSeries command and description
Work with WebSphere server jobs that run under the QEJBSVR user profile	`WRKUSRJOB USER(QEJBSVR) alternative you can use WRKSPLF USER(QEJBSVR)`
Delete spooled files for QEJBSVR user profile	`DLTSPLF FILE(*SELECT) SELECT(QEJBSVR)`

View the spooled files for a job	`WRKACTJOB SBS(QEJBAS5)`
	or
	`WRKACTJOB SBS(QEJBASND5)`
	1. Run one of the commands above.
	2. Enter option **8** (Work with Spooled Files) in the option line next to the active job whose spooled files you want to view.
	3. Enter option **5** (Display) in the option line next to the spooled file you want to view.
	Similarly, you can view ended jobs by using the Work with User Jobs (WRKUSRJOB) or Work with Job (WRKJOB) command.

WebSphere job logs

The job logs for the application server job may contain iSeries messages indicating the cause of a problem. The WebSphere Application Server product has several iSeries messages, which are located in QEJBAS5/QEJBMSGFS5. All of them are informational and range in severity from level 0 to 40.

The severity 0 messages are status messages for normal conditions, for example:

- EJB0105 Server starting with user profile &1 and JDK &2.
- EJB0106 WebSphere application server &1 ready.
- EJB0107 WebSphere application server ended.

The severity 30 and 40 messages are failure messages for abnormal conditions, for example:

- EJB0214 Unable to access application server configuration file.
- EJB0215 Value for application server property not valid.
- EJB6191 Application server already exists.
- EJB6192 Application server does not exist.
- EJB6193 Embedded JMS configuration creation failed.

The second level text for each message provides the details of the message and possible recovery (for severity 30 and 40 messages).

To display the messages, use either of the following commands:

```
DSPMSGD RANGE(message number) MSGF(QEJBAS5/QEJBMSGFA5)
DSPMSGD RANGE(*ALL) MSGF(QEJBAS5/QEJBMSGFA5)
```

The commands display information about the message. An example is shown in Figure 11-11.

```
                        Display Formatted Message Text
                                                                 System:   AS07
Message ID . . . . . . . . . :   EJB0004
Message file . . . . . . . . :   QEJBMSGFS5
  Library  . . . . . . . . . :     QEJBAS5

Message . . . . :   WebSphere application server not started.
Cause . . . . . :   The required Java Development Kit level could not be found
  for job &1. The WebSphere application server requires a JDK level of 1.3.
Recovery  . . . :   Install the supported JDK level and restart the server.




                                                                        Bottom
Press Enter to continue.

F3=Exit   F11=Display unformatted message text   F12=Cancel
```

Figure 11-11 The message file

If you are using an embedded JMS server, the jobs started on the QMQM subsystem can also have messages related to the JMS server. These messages are located in the QMQM/AMQMSG QMQMJAVA/QMQJMSG. All of them are informational and range in severity from level 0 to 40.

11.3.5 Monitoring WebSphere Application Server using the Log Analyzer Tool

The Log Analyzer merges all the data from one or more IBM Service logs and displays the entries. For more information on the IBM Service logs, see the IBM Service log file. Based on its symptom database, it analyzes and interprets the error conditions in the log entries to help you debug problems. The Log Analyzer can download the latest symptom database from the IBM Web site.

Obtaining the Log Analyzer

The Log Analyzer is installed on your workstation as part of the WebSphere Application Server workstation components. For details on how to install the workstation components, see Chapter 3, “Installation of workstation tools” on page 59.

Using the Log Analyzer

You can start the Log Analyzer on your client workstation by using any of these methods:

- From the Windows NT/2000 task bar, select **Start** -> **Programs** -> **IBM WebSphere** -> **Application Server Version 5.0** -> **Log Analyzer**.
- From the workstation command prompt:
  - On Windows NT or Windows 2000, enter this:

    ```
    [WAS_INSTALL_ROOT]\bin\waslogbr.bat
    ```

  - On UNIX platforms, enter this:

    ```
    [WAS_INSTALL_ROOT]/bin/waslogbr.sh
    ```

Once the Log Analyzer window is displayed, perform these steps to open an activity log file:

1. Do either of these steps:
   - Use FTP to send the activity.log file in binary format to your log analyzer workstation.
   - On Windows NT or Windows 2000, map a drive to the iSeries server. Alternatively, on UNIX platforms, mount a drive to the iSeries server.
2. Select **File -> Open**.
3. Navigate to the directory containing the activity.log file.
4. Select the activity.log file.
5. Select **Open**.
6. Expand the tree in the left-hand pane of the Log Analyzer to view messages.

Uncolored records are “normal”, yellow are warnings, and pink are errors. If you select a record, you see its contents, including the basic error or warning message, the date, the time, which WebSphere component logged the record, and which process (for example: application server, node agent) it came from, in the upper-right hand pane.

The activity log does not analyze any other log files, such as the SystemOut.log or native_stdout.log file. To analyze the records, right-click on a record in the tree on the left (click on the **UnitOfWorkView** at the top to get them all), and select **analyze**. See Figure 11-12 for a view example.

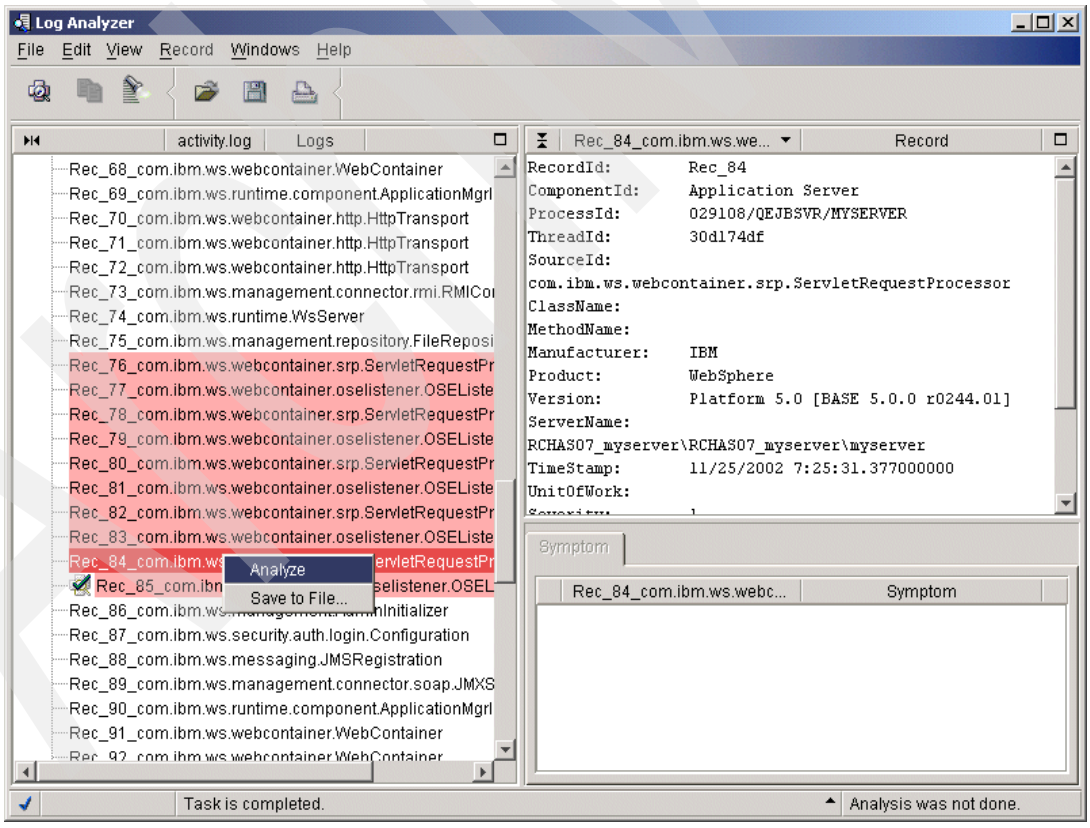


Figure 11-12 Example to analyze records by Log Analyzer

Any records with a green check mark next to them match a record in the symptom database. When you select a check-marked record, you'll see an explanation of the problem in the lower right-hand pane. See Figure 11-13 for an example.

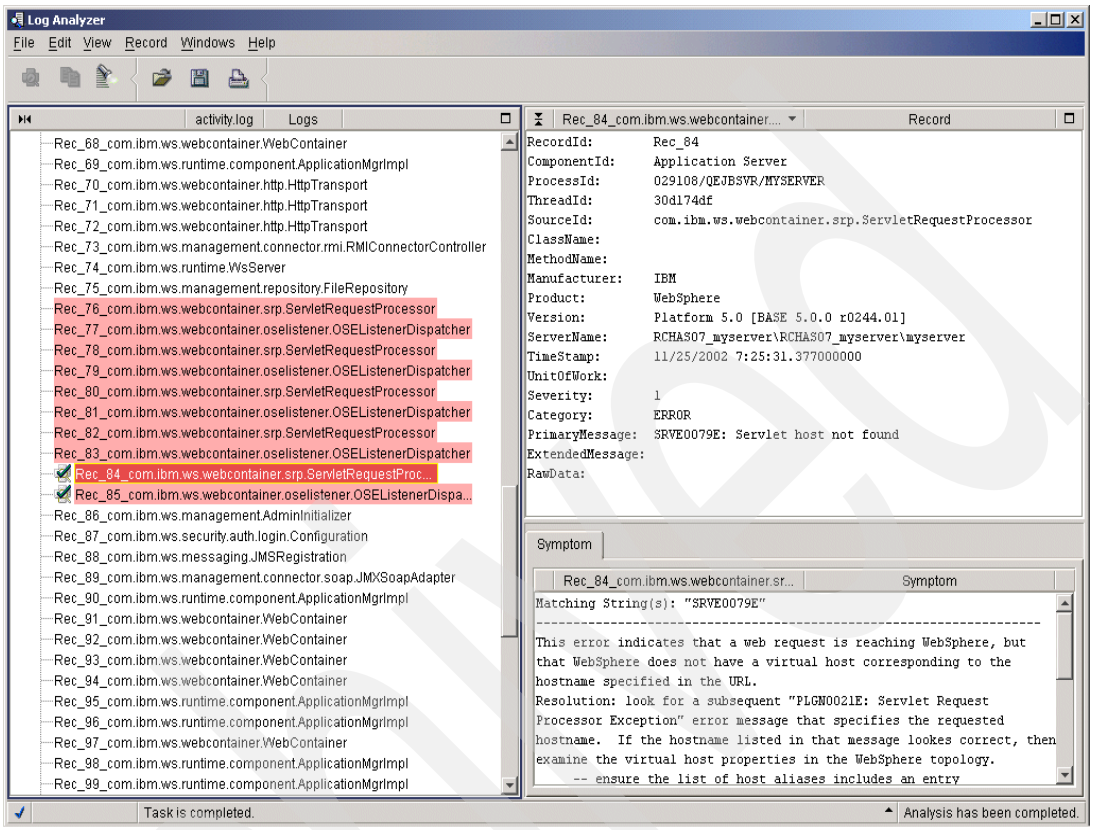


Figure 11-13 Example of results to analyze records

Use the Log Analyzer Help to get more detailed information on using Log Analyzer.

The Log Analyzer uses a symptom database to analyze the activity.log records. You can update this database from a URL page. That URL page is defined in the ivblogbr.properties file, as follows:

- On Windows NT or Windows 2000, enter this:

  `[WAS_INSTALL_ROOT]\properties\logbr\ivblogbr.properties`

- On UNIX platforms, enter this:

  `[WAS_INSTALL_ROOT]/properties/logbr/ivblogbr.properties`

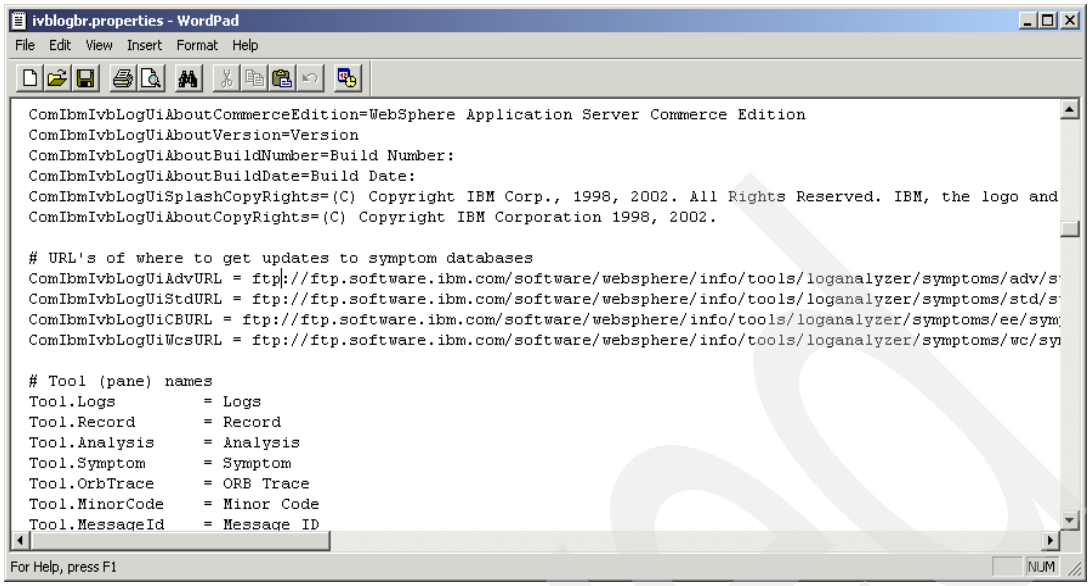


Figure 11-14 ivblogbr.properties file

Viewing the IBM Service log file without the Log Analyzer

If using the Log Analyzer to view the IBM Service log file is impractical or inconvenient, you can use an alternative tool named showlog. To do this, perform these steps:

1. On the iSeries command line, run the Start Qshell Interpreter (STRQSH) command.
2. Change directory to the bin directory for the product:
   `cd was_install_root/bin`
   Here, `was_install_root` is /QIBM/ProdData/WebAS5/Base for WebSphere Application Server or /QIBM/ProdData/WebAS5/ND for WebSphere Application Server Network Deployment.
3. Run the showlog script. The syntax is:
   `showlog [-instance instance_name] activity_log_file [output_file]`

   Here, the parameters are:

   -instance
   Optional. The value instance_name specifies the name of the WebSphere Application Server instance. The default value for instance_name is default.

   activity_log_file
   Required. The qualified integrated file system path name of the activity log file you wish to view. For example, `/QIBM/UserData/WebAS5/Base/myserver/logs/activity.log`

   output_file
   Optional. The qualified integrated file system path name of the file to contain the formatted, readable contents of the activity log file. If this parameter is not specified, the output is written to standard out (the screen).

   Examples:

```
showlog /QIBM/UserData/WebAS5/Base/<InstanceName>/logs/activity.log
showlog -instance myinst
/QIBM/UserData/WebAS5/ND/<InstanceName>/logs/activity.log
/home/myuserid/myinst_activity.txt
```

Here, <InstanceName> is the name of your WebSphere Application Server instance. The output of the showlog script file consists of message event entries separated by horizontal lines. Figure 11-15 shows an example of the output file.

```
Browse : /home/itscid16/myserver_log.txt
Record :    18  of   2197 by 18            Column :   1   133 by 131
Control :

....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+...
.9....+....0....+....1....+....2....+....3.
PrimaryMessage: ADMN0015I: AdminService initialized
ExtendedMessage:
---------------------------------------------------------------
ComponentId: Application Server
ProcessId: 029043/QEJBSVR/MYSERVER
ThreadId: f3a2afd0
SourceId: com.ibm.ws.security.auth.login.Configuration
ClassName:
MethodName:
Manufacturer: IBM
Product: WebSphere
Version: Platform 5.0 [BASE 5.0.0 r0244.01]
ServerName: RCHAS07_myserver\RCHAS07_myserver\myserver
TimeStamp: 2002-11-25 11:16:18.010000000
UnitOfWork:
Severity: 3
Category: AUDIT
PrimaryMessage:  SECJ0215I: Successfully set JAAS login provider configuration class to
com.ibm.ws.security.auth.login.Configuration
ExtendedMessage:

 F3=Exit   F10=Display Hex   F12=Cancel   F15=Services   F16=Repeat find   F19=Left
F20=Right
```

Figure 11-15 Example of the output file for showlog utility

Table 11-3 provides a description of the fields that make up the entries.

Table 11-3 Field description for showlog outfile

Field	Description
ComponentId	Currently not used
ProcessId	Job information for the job from which the message event was issued
ThreadId	Java (TM)thread ID of the thread from which the message event was issued.
FunctionName	Currently not used
ProbeId	Currently not used
SourceId	The class name of the object from which the message event was issued.
Manufacturer	IBM
Product	Product whose code issued the message event. Currently this is WebSphere Application Server.

Field	Description
ComponentId	Currently not used
ProcessId	Job information for the job from which the message event was issued
ThreadId	Java (TM)thread ID of the thread from which the message event was issued.
FunctionName	Currently not used
Version	Product version or build level
SOMProcessType	Currently not used
ServerName	
ClientHostName	Currently not used
ClientUserId	Currently not used
TimeStamp	Timestamp indicating when the message event was issued. The default time zone is Greenwich Mean Time (GMT).
UnitOfWork	Currently not used
Severity	Indicates the severity of the message. Audit message events are severity 3, warning message events are severity 2, and fatal/terminate/error message events are severity 1.
Category	The type of message event. Valid values are AUDIT, WARNING, ERROR, FATAL, and TERMINATE
FormatWarning	Currently not used
PrimaryMessage	Currently not used
ExtendedMessage	Message identifier. Message text.
RawDataLen	The length of any raw data that follows the message. Typically this value is zero.

11.3.6 WebSphere Application Server status messages

The administrative console includes a status message area that provides information on messages returned by the WebSphere Application Server regarding problems in your administrative configuration as well as messages about run-time events. For detailed information about the status message, refer to “WebSphere Application Server Message Reference” on page 458. Figure 11-16 shows an example of the status messages area.

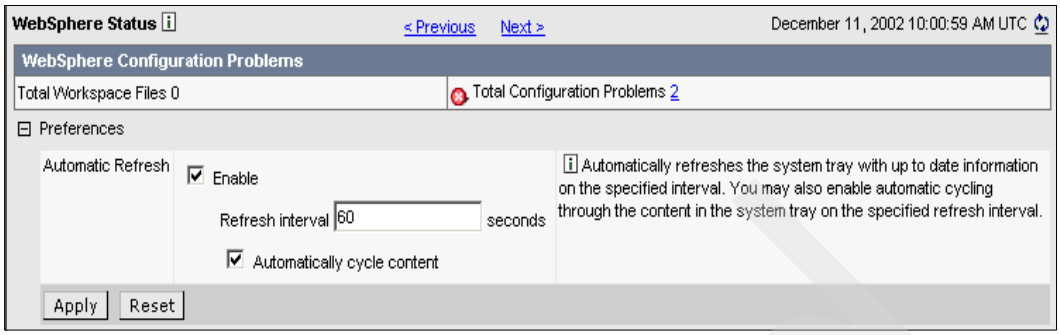


Figure 11-16 Status message area

You can click **Previous** or **Next** to toggle among displays of configuration problems and run-time events. Also, you can adjust the interval between automatic refreshes in the **Preferences** settings.

You can see a detailed description about each message logged into the status area. You must click on the kind of message to see all logged messages, and you can get more information about each particular message. Figure 11-17 shows an example of the detailed information.

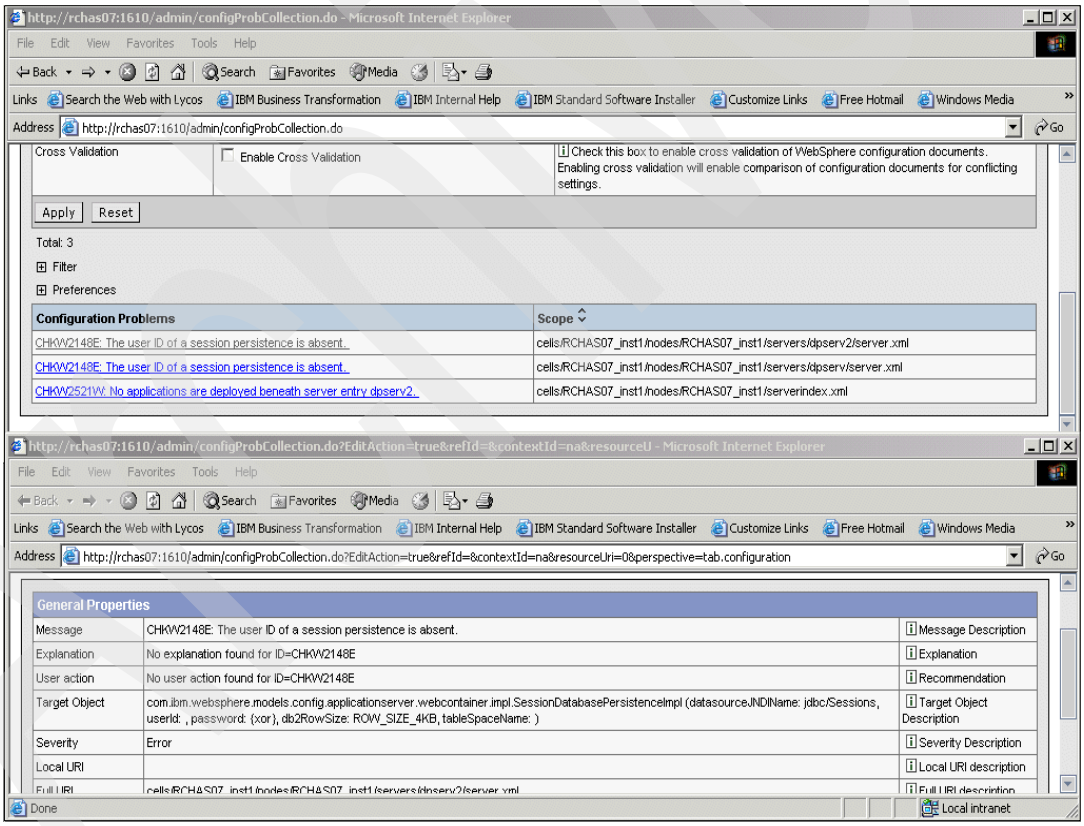


Figure 11-17 Status message detailed information

WebSphere Application Server Message Reference

You can log WebSphere Application Server system messages from a variety of sources, including application server components and applications. Messages logged by application server components and associated IBM products start with a unique message identifier that indicates the component or application that issued the message. The message identifier can be either 8 or 9 characters in length and has the form:

```
CCCC1234X
```

Here, `CCCC` is a four character alphabetic component or application identifier. `1234` is a four character numeric identifier used to identify the specific message for that component. `X` is an optional alphabetic severity indicator. (I=Informational, W=Warning, E=Error).

WebSphere Application Server for iSeries Infocenter includes a detailed description about message identifiers, explanation, and possible user action for messages issued by various WebSphere Application Server software components.If you want to see additional information about this topic refer to webSphere Application Server for iSeries Infocenter at:

```
http://publib.boulder.ibm.com/iseries/v1r1m0/websphere/ic2924/index.htm?info/rzaiz/50/
was.htm
```

11.3.7 Checking Product Activity logs and VLOG information

OS/400 tools, such as Start Service Tools (STRSST) and Work with Problems (WRKRPB), can help you to find additional information about possible problems on WebSphere Application Server and also on the others products installed on the iSeries system.

This kind of tools must be used by the iSeries administrators together with the IBM software support people. For additional support, you can contact your IBM local support, in accordance with your support contract.

The Work with Problems (WRKPRB) command

The Work with Problems command displays descriptions of the system problems, both system detected and user perceived. This command shows you both hardware and software problems detected in the system.

Sometimes this is a good tool for finding supplementary information about what is happening on your system. To see additional information about this command, refer to the **Troubleshooting and service** link in the iSeries Information Center at:

```
http://publib.boulder.ibm.com/pubs/html/as400/infocenter.html
```

Figure 11-18 shows an example of the WRKPRB command.

```
                              Work with Problems
                                                              System:   MYSYSTEM
Position to . . . . . .                   Problem ID

Type options, press Enter.
  2=Change   4=Delete   5=Display details    6=Print details
  8=Work with problem   9=Work with alerts   12=Enter text


Opt  Problem ID  Status         Problem Description
     0232236508  READY          Software problem data for QZHBWCFM has been lo
     0231763617  READY          Optical device OPT01 has detected a media or h
     0231763428  READY          Optical device OPT01 has detected a media or h
     0231763179  READY          Optical device OPT01 has detected a media or h
     0231762992  READY          Optical device OPT01 has detected a media or h
     0231762758  READY          Optical device OPT01 has detected a media or h
     0231762571  READY          Optical device OPT01 has detected a media or h
     0231239696  READY          *Attention*  Contact your hardware service pro


                                                                        Bottom
F3=Exit      F5=Refresh   F6=Print list     F11=Display dates and times
F12=Cancel   F16=Report prepared problems   F24=More keys
```

Figure 11-18 WRKPRB command

Start Service Tools (STRSST) command

Service Tools are special tools that help to find additional information about both hardware and software issues, and it is used by IBM support request. These tools have many options for making configurations in the iSeries system, and also contain the problems reported automatically by the system.

The Start Service Tool (STRSST) command asks for a Dedicated Service Tool (DST) user and password; you must contact the iSeries administrator to obtain this user profile. After that, you can see information related to system problems on two options: Product Activity Log (PAL) and Licensed Internal Code (LIC) Log.

Product Activity Log entries show information about hardware, licensed products, and licensed internal code problems. Each entry can be informational or can identify problems. Figure 11-19 shows an example of the PAL entries. To see this information, you must follow these steps:

1. Sign on the iSeries system using a user profile with *Service special authorization.
2. Type the **STRSST** command
3. Enter your DST user profile and password.
4. Select **Start Service Tool** option.
5. Select **Product Activity Log** option.
6. Select **Analyze log** option.
7. You can choose what kind of entries you want to see and also the period of time.

```
                          Log Analysis Report

From  . . :   11/01/02  10:20:48      To . . :   12/02/02  10:20:48

Type options, press Enter.
  5=Display report   6=Print report

     System                                        Resource      Resource
Opt  Ref Code     Date      Time      Class  Name          Type
     D6005501     11/20/02  07:55:17  Info
     B005A416     11/27/02  07:09:18  Temp   CMN08         2744
     B600FDC0     11/27/02  07:09:20  Temp   CMN08         2744
     B005A411     11/27/02  07:09:20  Temp   CMN08         2744
     B005A416     11/27/02  07:09:20  Temp   CMN08         2744
     B005A416     11/27/02  07:09:22  Temp   CMN08         2744
     B600FDC0     11/27/02  07:09:22  Temp   CMN08         2744

                                                                    Bottom
 F3=Exit
 F11=View Description                          F12=Cancel
```

Figure 11-19 PAL entries example

Licensed Internal Code entries show information about the Licensed Internal Code Log. Each entry has information about a specific LIC component which may have problems. For detailed information about the LIC entries, you can contact your IBM local support group. Figure 11-20 shows an example of the PAL entries. To see this information, you must follow these steps:

1. Sign on the iSeries system using a user profile with *Service special authorization.
2. Type **STRSST** command
3. Enter your DST user profile and password.
4. Select the option: **Start Service Tool**.
5. Select the option: **Licensed Internal Code log**.
6. Select the option: **Select entries from the Licensed Internal Code (LIC) log**.
7. You can choose what kind of entries you want to see and also the period of time.

```
                Select Entries from Licensed Internal Code Log

 Type options, press Enter to dump entry to selected device.
   1=Printer   2=Media   5=Display entry   8=Display note

      Entry     Major                     Major Minor                      Dump
 Opt  ID        Description               Code  Code  Date      Time     K bytes
      010022D8  Java virtual machine      4300  0731  11/13/02  16:36:12       1
010023A8  Recovered object list    0405  01A1  11/20/02  07:56:34       0
      010023AA  Commit                    0F00  1801  11/20/02  07:56:36       1
      010023AB  Common function           1B00  0001  11/20/02  07:56:39       1
      010023AC  IPL                       0500  0000  11/20/02  07:56:40       1
      010023AD  IPL                       0500  ADFE  11/20/02  07:58:31      23
      010023AE  Error analysis            4A00  3517  11/20/02  07:59:01       1
      010023B0  Tolerance                 0400  0000  11/20/02  11:31:00       0
      010023B2  Database                  0600  5CFC  11/21/02  00:00:45       1
      010023B3  Database                  0600  5CFC  11/21/02  00:00:46       1
      010023B4  Synchronization           2100  0140  11/22/02  11:26:09       0
                                                                           Bottom
 F3=Exit   F12=Cancel
```

Figure 11-20 LIC log entries example

11.4 WebSphere Application Server log files

WebSphere Application Server has a variety of logs to which messages are written. For example, system messages, which can be written by any application server component or application are written to general-purposes logs such as the JVM logs and the IBM service, or activity log. Other logs are very specific in nature and are scope to a particular component or activity. For example, the HTTP server plugin maintains a component-specific plugin log.

In general, the general purpose logs such as the JVM logs and the IBM service, or the activity log, are used to monitor the health of the application server and are used in most problem determination procedures. However, the problem determination procedure for a specific component may direct you to examine the contents of a component or product specific log.

The topics below describe the log files available for WebSphere Application Server, and how you can configure and view the files.

11.4.1 JVM log files

The JVM log files are one of the first places to start when troubleshooting a problem. These log files contain the output for the System.out and System.err output streams for the application server process. There is one log file specified for the System.out output stream and one file specified for the System.err output stream. The JVM logs contain print data written by applications. The data may be written directly by the application in the form of System.out.print(), System.err.print(), etc., method calls. Data may also be written indirectly by the application calling a JVM function, such as Exception.printStackTrace(). In addition, the System.out JVM log contains system messages (also known as message events) written by the WebSphere application server.

The JVM log files are self-managing in that you can configure the files to not grow beyond a certain size, and you can configure the number of historical, or archived, files to retain. In addition, you can configure the log files to rollover (be archived) based on time as well as size.

The granted authorities for the files are shown in Table 11-4.

Table 11-4 Authorities to the log files

User profile	Access
*PUBLIC	*EXCLUDE
QEJBSVR	*RW

If your application server is running under a user profile other than the default (QEJBSVR) and that user profile does not have QEJBSVR specified as a group profile, you must explicitly grant *RW authority to the user profile for the activity.log file.

Depending on how the JVM log is configured, application print data may be formatted to look like WebSphere system messages, or may be displayed as plain text with no additional formatting. WebSphere system messages are always formatted.

Configuring the JVM log files

Use the administrative console to configure the JVM logs for an application server. Configuration changes for the JVM logs that are made to a running application server are not applied until the next restart of the application server. Figure 11-21 shows an example of the JVM logs configuration.

Perform these steps to configure the JVM logs:

1. Start the administrative console.
2. Expand **Troubleshooting** and click **Logs and Trace**.
3. Click the link for the server you wish to configure.
4. Click **JVM Logs**.
5. Select the **Configuration** tab.
6. Scroll through the panel to display the attributes for the stream to be configured. You can change these attributes according to your needs. These are the available attributes:
   - File Name: The name of the System.out or System.err file.
   - File formatting: You can choose between **Basic** and **Advanced** format. For more information about messages format, refer to “Message formats” on page 465.
   - Log File rotation. The JVM logs are self-managing log files and you can define the criteria to the system create another file (file size or time).
   - Maximum Number of historical log files: Specifies the number of historical, or rolled over, files to keep.
   - Show application print statements: Causes application messages written to this stream using the print and println stream methods to be shown.
   - Format print statements: Causes application messages written to this stream using the print and println stream methods to be formatted like WebSphere system messages.
7. Click **Apply**.
8. Click the **Save** link at the top of the page to save your configuration changes. Click **Save** on the resulting page.

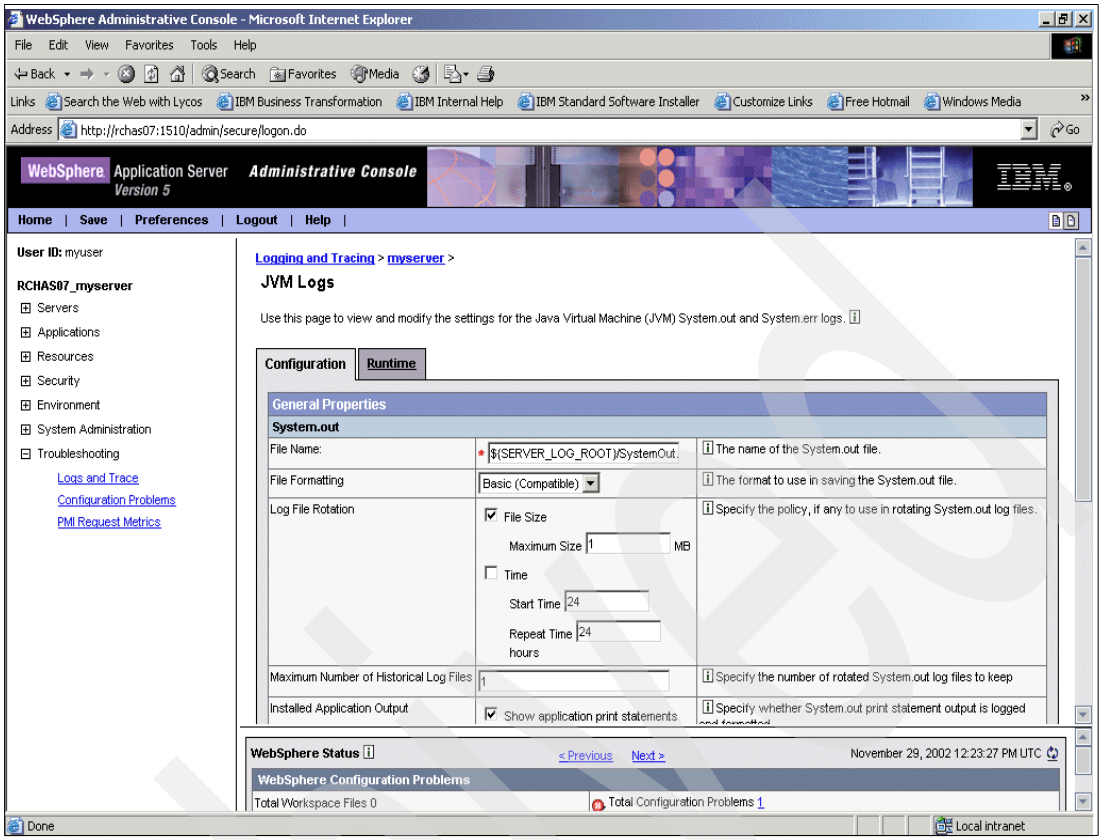


Figure 11-21 Example of JVM logs configuration

Viewing the JVM log files

The JVM logs are written as plain ASCII text files. By default, the JVM logs are located in the `logs/<ServerName>` subdirectory of the WebSphere instance you are using, where `<ServerName>` is the name of the server. If you are using the default WebSphere Application Server instance, the path is /QIBM/UserData/WebAS5/Base/default/logs/server1.

If you are using the default Network Deployment instance, the path is /QIBM/UserData/WebAS5/ND/default/logs/dmgr. For a WebSphere Application Server instance that has been added to a Network Deployment domain (cell), the log files for the node agent are located in subdirectory logs/nodeagent and the log files for the JMS server are located in subdirectory logs/jmsserver.

You can view the JVM log files using one of these methods:

- View the JVM logs from the administrative console.
  Figure 11-22 shows an example of the view option from the administrative console. Perform these steps to view the JVM logs using the administrative console:
  a. Start the administrative console.
  b. Expand **Troubleshooting**.
  c. Click **Logs and Trace**.
  d. Click the link for the server whose logs you wish to view.
  e. Click **JVM Logs**.
  f. Select the **Runtime** tab.
  g. Click **View** for the log you want to view.

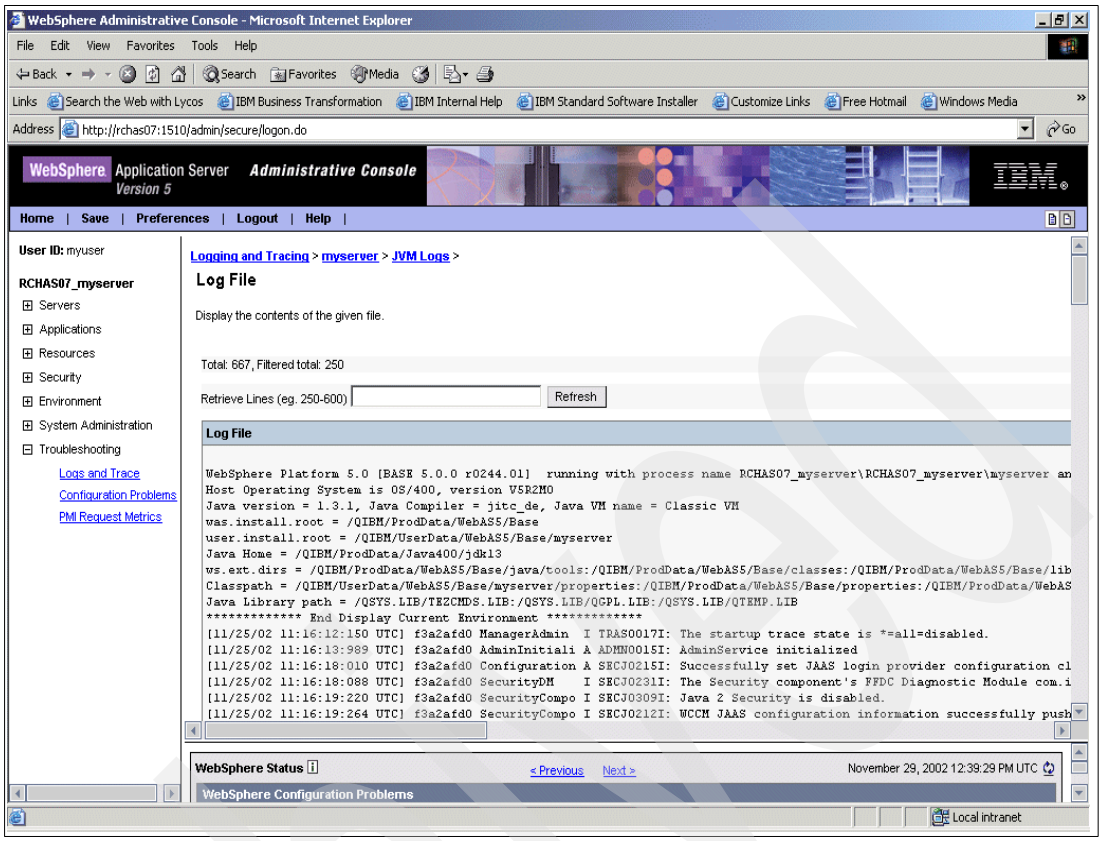


Figure 11-22 Example View JVM logs by Administrative Console

- View the JVM logs on the iSeries where they are stored.
  Figure 11-23 shows an example of the view option from the EDTF command.

  To use the Edit File (EDTF) CL command to view the JVM log files:
  From the OS/400 command line, invoke the EDTF CL command specifying the Integrated File System pathname for the file you wish to view. For example:

```
EDTF STMF('/QIBM/UserData/WebAS5/Base/MyServer/logs/MyServer/SystemOut.log')
EDTF STMF('/QIBM/UserData/WebAS5/Base/MyServer/logs/MyServer/SystemErr.log')
```

```
 Edit File: /QIBM/UserData/WebAS5/Base/myserver/logs/myserver/systemout.log
 Record :      1   of    668 by  10                  Column :    1    353 by 126
 Control :

CMD
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9....+....0....+....1....
+....2....+..
    ************Beginning of data**************
   ************ Start Display Current Environment ************
    WebSphere Platform 5.0 [BASE 5.0.0 r0244.01]  running with process name
RCHAS07_myserver\RCHAS07_myserver\myserver and process
   Host Operating System is OS/400, version V5R2M0
   Java version = 1.3.1, Java Compiler = jitc_de, Java VM name = Classic VM
   was.install.root = /QIBM/ProdData/WebAS5/Base
   user.install.root = /QIBM/UserData/WebAS5/Base/myserver
   Java Home = /QIBM/ProdData/Java400/jdk13
    ws.ext.dirs =
/QIBM/ProdData/WebAS5/Base/java/tools:/QIBM/ProdData/WebAS5/Base/classes:/QIBM/ProdData/WebAS5/Base/lib/server:/Q
    Classpath =
/QIBM/UserData/WebAS5/Base/myserver/properties:/QIBM/ProdData/WebAS5/Base/properties:/QIBM/ProdData/WebAS5/Base/li
b
   Java Library path = /QSYS.LIB/TEZCMDS.LIB:/QSYS.LIB/QGPL.LIB:/QSYS.LIB/QTEMP.LIB
   ************* End Display Current Environment *************
  [11/25/02 11:16:12:150 UTC] f3a2afd0 ManagerAdmin  I TRAS0017I: The startup trace state is *=all=disabled.
   [11/25/02 11:16:13:989 UTC] f3a2afd0 AdminInitiali A ADMN0015I: AdminService initialized
    [11/25/02 11:16:18:010 UTC] f3a2afd0 Configuration A SECJ0215I: Successfully set JAAS login provider
configuration class to com
    [11/25/02 11:16:18:088 UTC] f3a2afd0 SecurityDM    I SECJ0231I: The Security component's FFDC Diagnostic
Module com.ibm.ws.secu
   [11/25/02 11:16:19:220 UTC] f3a2afd0 SecurityCompo I SECJ0309I: Java 2 Security is disabled.
    [11/25/02 11:16:19:264 UTC] f3a2afd0 SecurityCompo I SECJ0212I: WCCM JAAS configuration information
successfully pushed to logi


 F2=Save  F3=Save/Exit  F12=Exit  F15=Services  F16=Repeat find  F17=Repeat change  F19=Left  F20=Right
```

Figure 11-23 Example view JVM logs by EDTF

- From a non-iSeries workstation.
  Perform these steps to view the JVM logs from a mapped or mounted driver:
  a. Map (Windows 32-bit workstation) or mount (UNIX workstation) a drive to the iSeries.
  b. Open the file in a text editor or drag and drop the file into a file editing/viewing program.

Interpreting the JVM log files

The JVM logs contain print data written by applications. The data may be written directly by the application in the form of System.out.print(), System.err.print(), etc., method calls. Data may also be written indirectly by the application calling a JVM function, such as Exception.printStackTrace(). In addition, the System.out JVM log contains system messages written by the WebSphere application server.

Depending on how the JVM log is configured, application print data may be formatted to look like WebSphere system messages, or may be displayed as plain text with no additional formatting. WebSphere system messages are always formatted.

Formatted messages may be written to the JVM logs in either basic or advanced format, depending on how the JVM log is configured.

This information describes the two formats and the fields that make up the messages.

Message formats

Formatted messages may be written to the JVM logs in one of two formats:

- Basic
- Advanced

Basic Format: This is the format used in earlier versions of WebSphere application server. Message events displayed in basic format use this format. The notation <name> indicates mandatory fields that are always displayed in the basic format message. The notation [name]indicates optional or conditional fields that are included if they can be determined.

```
TimeStampThreadIdShortNameEventType[ClassName][MethodName]message
```

Advanced Format: This extends the basic format by adding information about an event, when possible.

Message events displayed in advanced format use this format. The notation <name> is used to indicate mandatory fields that are always displayed in the advanced format for message entries. The notation [name] is used to indicate optional or conditional fields that are included if they can be determined.

```
TimeStampThreadIdEventTypeUOWsource=LongName[ClassName]
[methodName]OrganizationProductComponentmessage
```

Basic and advanced format fields

Basic and Advanced Formats use many of the same fields and formatting techniques. Figure 11-24 shows an example of the format field. The various fields that may be found in these formats include:

- **TimeStamp:** The timestamp is formatted using the locale of the process where it is formatted. It includes a fully qualified date (for example YYMMDD), 24 hour time with millisecond precision and a time zone.
- **Thread:** An 8-character hexidecimal value generated from the hash code of the thread that issued the message.
- **ShortName:** The abbreviated name of the logging component that issued the message or trace event. This is typically the class name for WebSphere internal components, but may be some other identifier for user applications.
- **LongName:** The full name of the logging component that issued the message or trace event. This is typically the fully qualified class name for WebSphere internal components, but may be some other identifier for user applications.
- **EventType:** A 1-character field that indicates the type of the message or trace event. Message types are in upper case. Possible values include:
  - **A**: An Audit message.
  - **I**: An Informational message.
  - **W**: A Warning message.
  - **E**: An Error message.
  - **F**: A Fatal message.
  - **O**: A message that was written directly to System.out by the user application or WebSphere internal components.
  - **R**: A message that was written directly to System.err by the user application or WebSphere internal components.
  - **U**: A special message type used by the message logging component of the WebSphere run time.
  - **Z**: A placeholder to indicate the type was not recognized.
- **ClassName:** The class that issued the message or trace event.
- **MethodName:** The method that issued the message or trace event.
- **Organization:** The organization that owns the application that issued the message or trace event.
- **Product:** The product that issued the message or trace event.

- **Component:** The component within the product that issued the message or trace event.
- **UOW:** The unit of work identifier for the event. This field is not currently used.

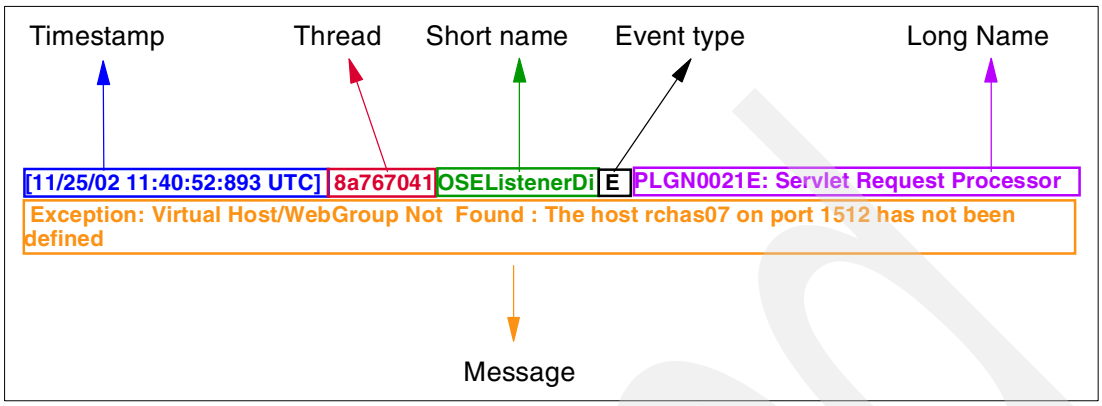


Figure 11-24 Example of the message format

11.4.2 Process log files

Application server processes contain two output streams that are accessible to native code running in the process. These streams are the stdout and stderr streams. Native code, including the JVM, may write data to these process streams.

By default, the stdout and stderr streams are redirected to log files at application server startup, which contain text written to the stdout and stderr streams by native modules (for example, *SRVPGM objects). The WebSphere Application Server does no special processing or formatting of the output that is written to the process logs.

This is a change from previous versions of WebSphere, which by default had one log file for both JVM standard output and native standard output, and one log file for both JVM standard error and native error output.

The granted authorities for the files are shown in Table 11-5.

Table 11-5 Authorities to the log files

User profile	Access
*PUBLIC	*EXCLUDE
QEJBSVR	*RX

If your application server is running under a user profile other than the default (QEJBSVR) and that user profile does not have QEJBSVR specified as a group profile, you must explicitly grant *RW authority to the user profile for the activity.log file.

Configuring the process log files

Use the administrative console to configure the process log files for an application server. The file name for a process log is the only attribute that can be changed. Unlike the JVM logs, the process logs are not self-managing so the size and number of historical files cannot be configured. Generally, these file are empty or contain very small amounts of information, so self-management is not required. Figure 11-25 shows an example of the process logs configuration.

Configuration changes for the process logs that are made to a running application server are not applied until the next restart of the application server.

Perform these steps to configure the process logs:

1. Start the administrative console.
2. Expand **Troubleshooting** and click **Logs and Trace**.
3. Click the link for the server you wish to configure.
4. Click **Process Logs**.
5. Select the **Configuration** tab.
6. The file name for each of the process streams (stdout and stderr) is displayed.
7. Change the file names as appropriate.
8. Click **Apply**.
9. Click the **Save** link at the top of the page to save your configuration changes. Click **Save** on the resulting page.

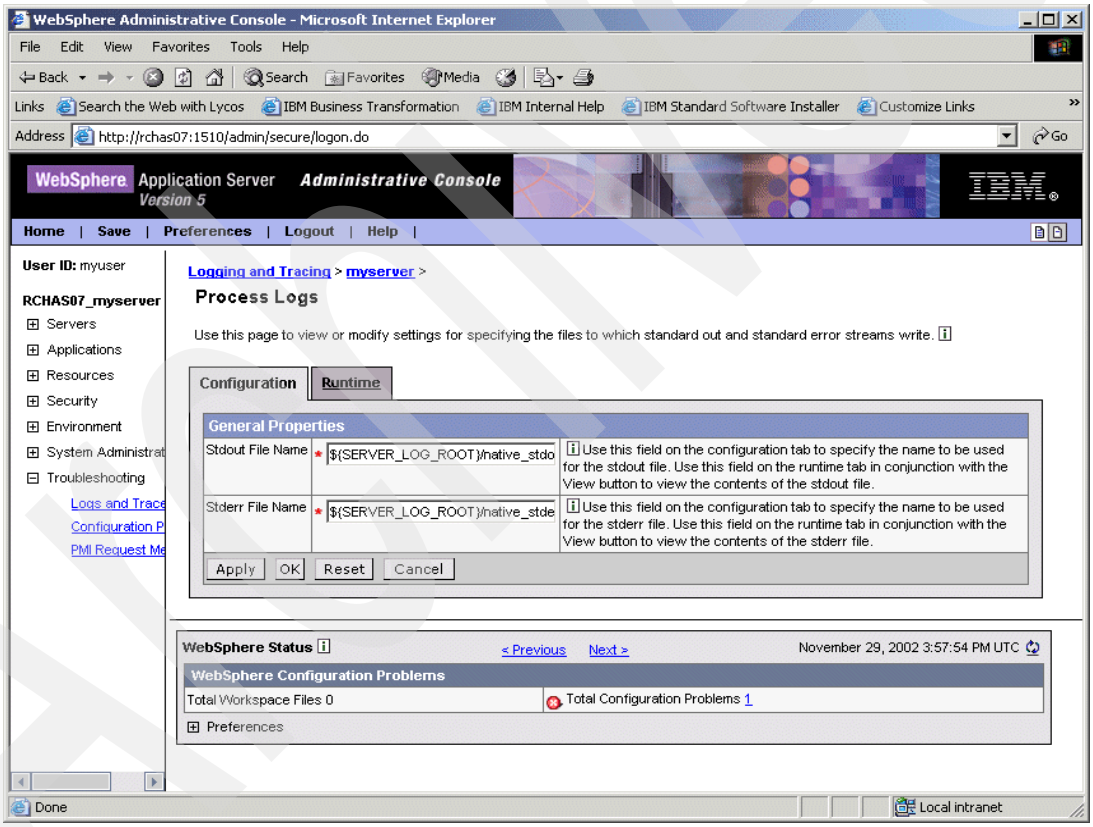


Figure 11-25 Example process log files configuration

Viewing the process log files

The process logs are written as plain ASCII text files. By default, the process logs are located in the logs/servername subdirectory of the WebSphere instance you are using, where servername is the name of the server. If you are using the default WebSphere Application Server instance, the path is /QIBM/UserData/WebAS5/Base/default/logs/server1.

If you are using the default Network Deployment instance, the path is /QIBM/UserData/WebAS5/ND/default/logs/dmgr. For a WebSphere Application Server

instance that has been added to a Network Deployment domain (cell), the log files for the node agent are located in subdirectory logs/nodeagent, and the log files for the JMS server are located in subdirectory logs/jmsserver.

You can view the process log files using one of these methods:

- View the process logs from the administrative console.
  Figure 11-26 shows an example of the view option from the administrative console. Perform these steps to view the process logs using the administrative console:
  a. Start the administrative console.
  b. Expand **Troubleshooting**.
  c. Click **Logs and Trace**.
  d. Click the link for the server whose logs you wish to view.
  e. Click **Process Logs**.
  f. Select the **Runtime** tab.
  g. Click **View** for the log you want to view.

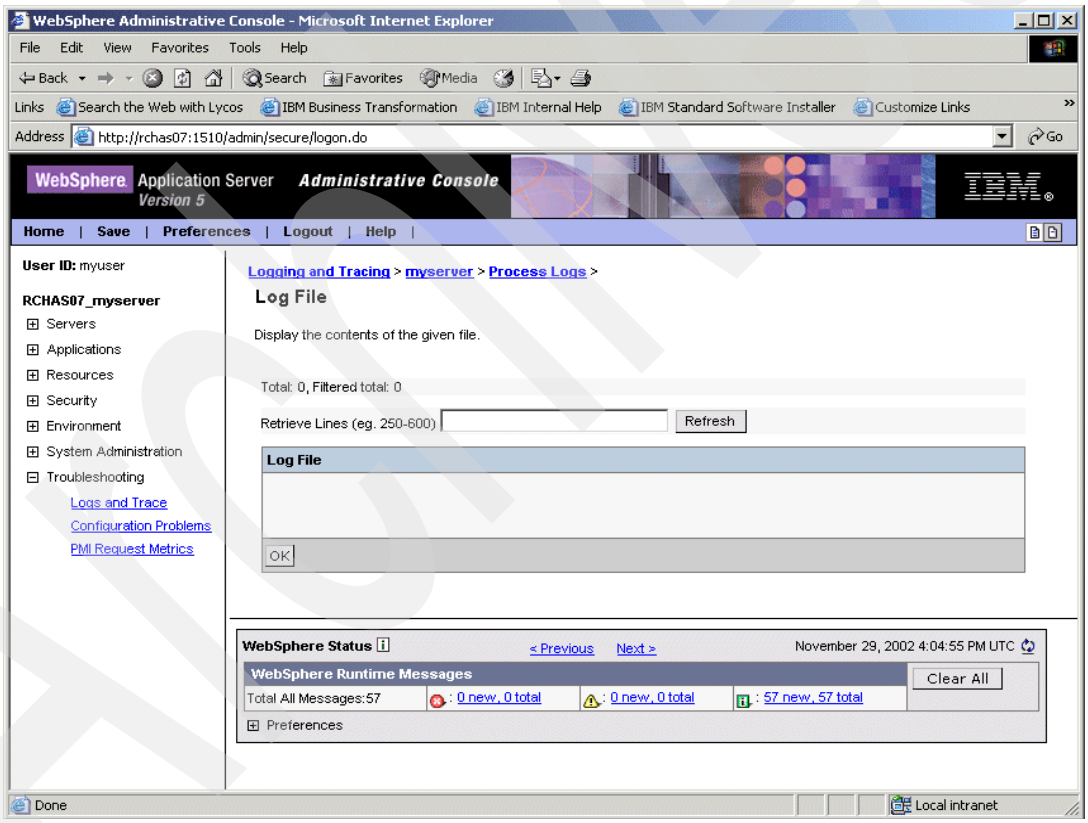


Figure 11-26 Example View process logs by administrative console

- View the process logs on the iSeries where they are stored (Figure 11-27).

  You can use the Edit File (EDTF) CL command to view the process log files. From the OS/400 command line, invoke the EDTF CL command specifying the Integrated File System pathname for the file you wish to view. For example:

```
EDTF STMF('/QIBM/UserData/WebAS5/Base/default/logs/server1/native_stdout.log')
EDTF STMF('/QIBM/UserData/WebAS5/Base/default/logs/server1/native_stderr.log')
```

```
 Edit File: /QIBM/UserData/WebAS5/Base/myserver/logs/myserver/native_stdout.log
 Record :       1   of       2 by  10                      Column :    1     72 by 126
 Control :

CMD ....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....
     ************Beginning of data**************
    Attaching Java program to /QIBM/ProdData/mqm/java/lib/com.ibm.mq.JAR.
    Attaching Java program to /QIBM/ProdData/mqm/java/lib/com.ibm.mqjms.JAR.
     ************End of Data********************
```

Figure 11-27 Example view process logs by EDTF

- From a non-iSeries workstation.
  Perform these steps to view the process logs from a mapped or mounted driver:
  a. Map (Windows 32-bit workstation) or mount (UNIX workstation) a drive to the iSeries.
  b. Open the file in a text editor or drag and drop the file into a file editing/viewing program.

11.4.3 IBM Service log

The IBM Service, or activity, log file is a binary file to which WebSphere Application Server writes the message events for any servers running under an instance (node). The WebSphere Application Server runtime creates the file (named activity.log by default) in the logs subdirectory of your WebSphere Application Server instance. For the default WebSphere Application Server instance, this subdirectory is /QIBM/UserData/WebAS5/Base/default/logs. For the default WebSphere Application Server Network Deployment instance, this subdirectory is /QIBM/UserData/WebAS5/ND/default/logs. The granted authorities for the files are shown in Table 11-6.

Table 11-6 Authorities to the log files

User profile	Access
*PUBLIC	*EXCLUDE
QEJBSVR	*RW

If your application server is running under a user profile other than the default (QEJBSVR) and that user profile does not have QEJBSVR specified as a group profile, you must explicitly grant *RW authority to the user profile for the activity.log file.

Configuring the IBM Service log file

Use the administrative console to configure the IBM Service log for an application server. Configuration changes for the activity log that are made to a running application server are not applied until the next restart of the application server. Figure 11-28 shows a configuration example of IBM service log file.

Perform these steps to configure the IBM Service log:

1. Start the administrative console.
2. Expand **Troubleshooting** and click **Logs and Trace**.
3. Click the link for the server you wish to configure.
4. Click **IBM Service Logs**.
5. Select the **Configuration** tab.

6. Scroll through the panel to display the attributes for the stream to be configured. You can change these attributes according to your needs. These are the available attributes:
   - Enable service log. Check this box to enable the creation of a service log file to store data created by the IBM Service logs.
   - File Name. Specifies the name of the service log for the application server.
   - Maximum File Size. Specifies the maximum size in megabytes of the service log file.
   - Message Filtering. Specifies what classes of messages will be stored in the service log. You can choose between:
     - Log all messages
     - Log warning, error
     - Log service, warning, error
     - Log error
   - Enable Correlation ID. You can use the correlation ID to correlate activity to a particular client request, or correlate activities on multiple application servers.
7. Click the **Apply** button.
8. Click the **Save** link at the top of the page to save your configuration changes. Click **Save** on the resulting page.

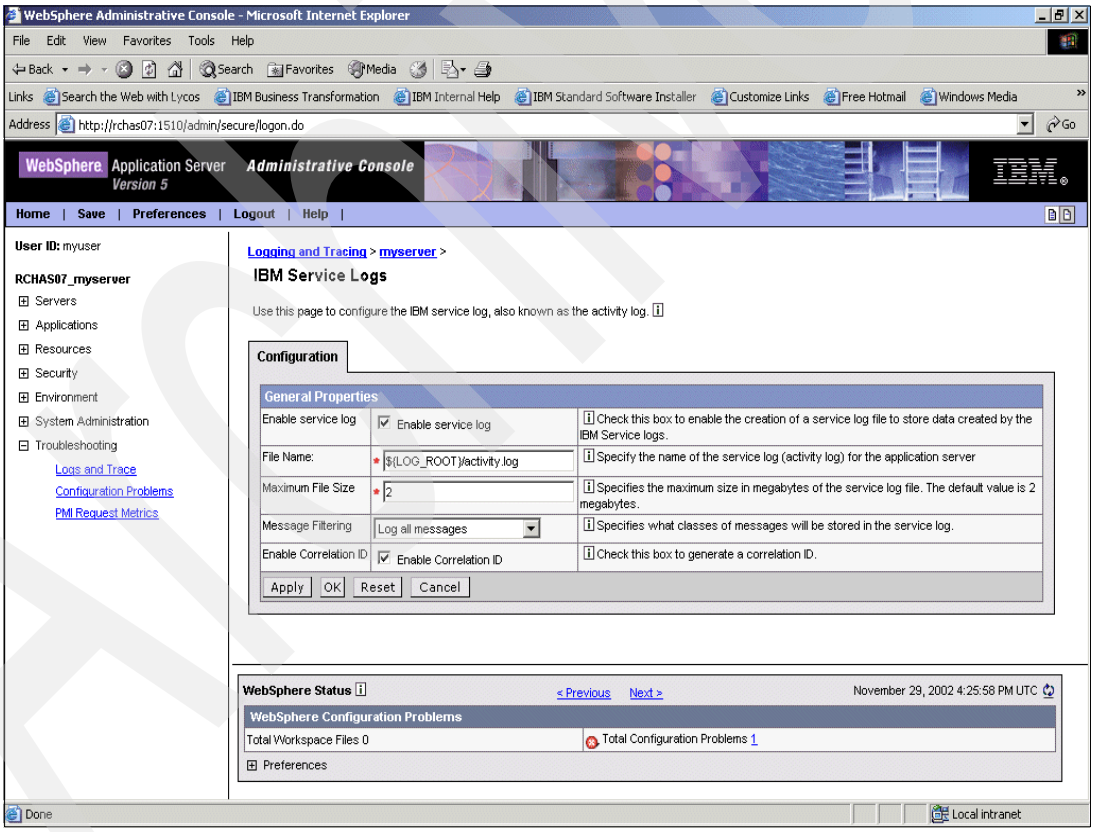


Figure 11-28 Example of IBM service log configuration

Viewing the IBM service log file

For information about how you can view the contents of the IBM service log, refer to 11.3.5, “Monitoring WebSphere Application Server using the Log Analyzer Tool” on page 451.

11.5 Tracing

Tracing is useful when you are experiencing a problem whose cause cannot be determined from the WebSphere Application Server log files or other normal problem determination channels. Trace allows you to obtain detailed information about the execution of WebSphere Application Server components, including application servers, clients, and other processes in the environment. Trace files show the time and sequence of methods called by WebSphere Application Server runtime classes, and can be used to pinpoint the failure. Generally, you should not require the use of trace files to determine what is causing an application problem; however, IBM Support personnel may request that you provide traces for a problem.

11.5.1 Enabling the trace service

You can use the administrative console to enable the trace service for an application server. The trace service can be enabled dynamically or statically for an application server.

When you enable trace dynamically for a running application server, the trace settings are in effect only for the lifetime of the server. The trace settings are not saved to the application server configuration. Use dynamic trace when the problem you are diagnosing occurs after the application server has started successfully.

When you enable trace statically for an application server, the trace settings are not enabled until the application server is started (or restarted). The trace settings are used every time you start the application server. Use static trace when the problem you are diagnosing occurs during application server startup.

Note: Depending on the amount of trace event data being collected, the trace service can negatively affect performance of the application server. Once you have gathered the appropriate trace, be sure to disable the trace service.

Enabling the trace service

Perform these steps to enable the trace service:

1. Start the administrative console.
2. Expand **Troubleshooting** and click **Logs and Trace**.
3. Click the link for the server you wish to configure.
4. Click **Diagnostic trace**. To enable trace statically, select the **Configuration** tab.
   To enable trace dynamically, select the **Runtime** tab.
   Figure 11-29 shows an example for enabling a dynamic trace from the administrative console.

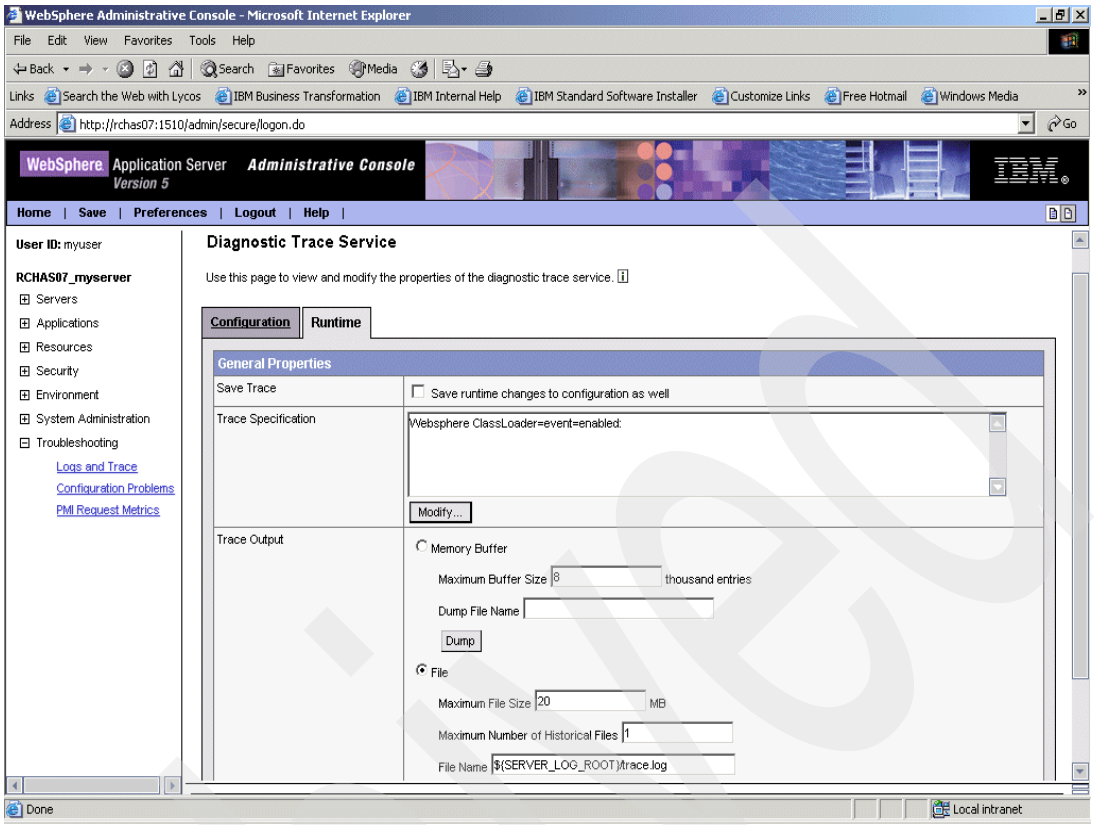


Figure 11-29 Example for enabling dynamic trace by administrative console

5. Scroll through the panel to display the current or default trace settings.
6. Change the appropriate configuration attributes. You can specify these attributes:
   a. Trace specification. You can select all for tracing all WebSphere Application Server components, or pick Modify button to select the component that you want to trace. Figure 11-30 shows an example of the available groups that you can use on your trace. When you define the component to trace, you must select the level of tracing, which could be:
      - All disabled
      - Entry/exit
      - Event
      - Debug
      - Entry/exit + event
      - Entry/exit + debug
      - Event + debug
      - All enabled

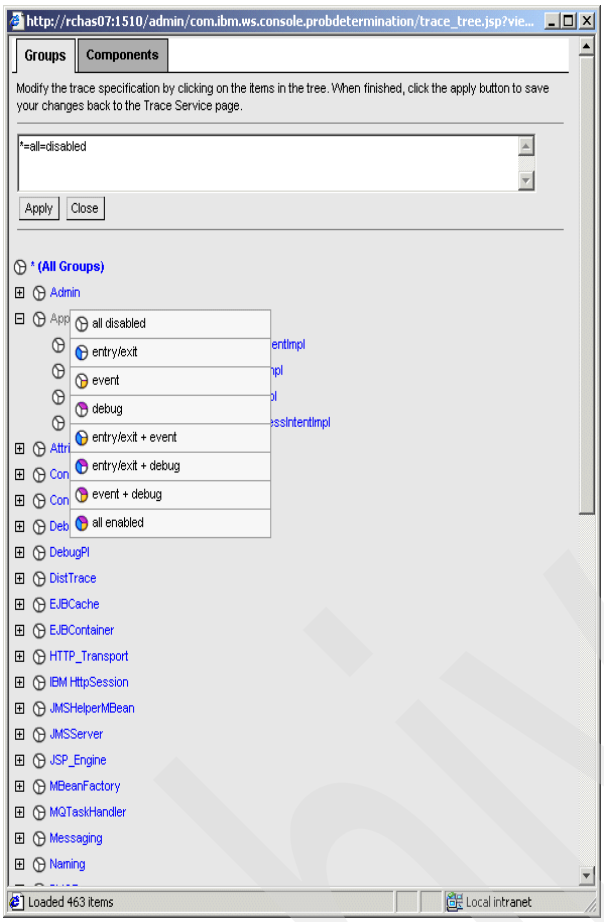


Figure 11-30 Groups available to trace

b. Trace output. You can choose between:

- Memory buffer. You must define the quantity of entries that memory buffer will have.
- File. You must define the maximum file size and the file name for the trace.

c. Trace output format. This is only available to static trace. You can choose between:

- Basic
- Advanced
- Log Analyzer

7. Click **Apply**.
8. If you are enabling static trace, click the **Save** link at the top of the page to save your configuration changes. Click **Save** on the resulting page.

11.5.2 Disabling the trace service

Figure 11-31 shows an example for enabling a dynamic trace from the administrative console. Perform these steps to disable the trace service:

1. Start the administrative console.
2. Expand **Troubleshooting** and click **Logs and Trace**.
3. Click the link for the server you wish to configure.
4. Click **Diagnostic trace**.

5. To disable static trace, select the **Configuration** tab.
   Uncheck the **Enable Trace** checkbox.
   To disable dynamic trace, select the **Runtime** tab.
   Change the **Trace Specification** to *=all=disabled.
6. Click **Apply**.
7. If you are disabling **static** trace, click the **Save** link at the top of the page to save your configuration changes. Click **Save** on the resulting page.

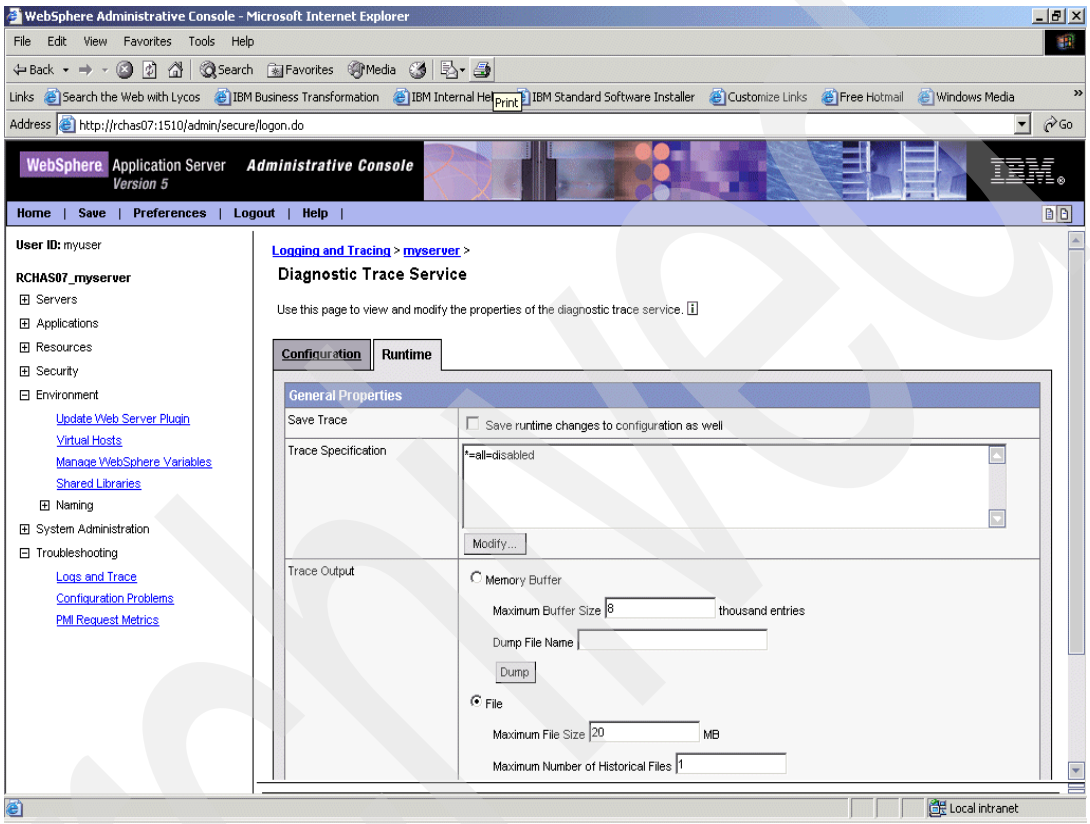


Figure 11-31 Example for disabling dynamic trace by administrative console

11.5.3 Interpreting the Trace Service output

On an application server, trace output can be directed either to a file or to an in-memory circular buffer. If trace output is directed to the in-memory circular buffer, it must be dumped to a file before it can be viewed.

On an application client or standalone process, trace output can be directed either to a file or to the process console window.

In all cases, trace output is generated as plain text in either basic, advanced, or log analyzer format, as specified by the user when enabling the trace service.

In the following sections we describe the three formats and the fields that make up the messages.

Trace output formats

Formatted trace events may be written to the trace file in one of three formats:

- **Basic Format:** This is the format used in earlier versions of WebSphere application server.
- **Advanced Format:** This extends the basic format by adding information about an event, when possible.
- **Log Analyzer Format:** This is the same binary format used for the IBM Service (activity.log) log file. This format allows you or IBM Service to use the Log Analyzer tool to open and interpret the trace output. For additional information about the Log Analyzer tool, you can see “Using the Log Analyzer” on page 451.

Basic and advanced format fields

Basic and Advanced Formats use many of the same fields and formatting techniques. The various fields that may be found in these formats include:

- **TimeStamp:** The timestamp is formatted using the locale of the process where it is formatted. It includes a fully qualified date (for example YYMMDD), 24 hour time with millisecond precision and a time zone.
- **Thread:** An 8-character hexidecimal value generated from the hash code of the thread that issued the message.
- **ShortName:** The abbreviated name of the logging component that issued the message or trace event. This is typically the class name for WebSphere internal components, but may be some other identifier for user applications.
- **LongName:** The full name of the logging component that issued the message or trace event. This is typically the fully qualified class name for WebSphere internal components, but may be some other identifier for user applications.
- **EventType:** A 1-character field that indicates the type of the message or trace event. Message types are in upper case. Possible values include:
  - **A**: An Audit message.
  - **I**: An Informational message.
  - **W**: A Warning message.
  - **E**: An Error message.
  - **F**: A Fatal message.
  - **O**: A message that was written directly to System.out by the user application or WebSphere internal components.
  - **R**: A message that was written directly to System.err by the user application or WebSphere internal components.
  - **U**: A special message type used by the message logging component of the WebSphere run time.
  - **Z**: A placeholder to indicate the type was not recognized.
- **ClassName:** The class that issued the message or trace event.
- **MethodName:** The method that issued the message or trace event.
- **Organization:** The organization that owns the application that issued the message or trace event.
- **Product:** The product that issued the message or trace event.
- **Component:** The component within the product that issued the message or trace event.
- **UOW:** The unit of work identifier for the event. This field is not currently used.

Basic format

Trace events displayed in basic format use this format, where *name* indicates mandatory fields that are always displayed in the formatted message, and [name] indicates optional fields that are displayed if they can be determined.

```
TimeStampThreadIdShortNameEventType[ClassName][MethodName]
textmessage[parameter 1][parameter 2]
```

Advanced format

Trace events displayed in advanced format use this format, where *name* indicates mandatory fields that are always displayed in the formatted message, and [name] indicates optional fields that are displayed if they can be determined.

```
TimeStampThreadIdEventTypeUOWsource=LongName[ClassName][MethodName]
OrganizationProductComponenttextMessage[parameter 1=parameterValue]
[parameter 2=parameterValue]
```

11.6 Collecting data to calling support

If you are not able to resolve a WebSphere Application server problem by following the steps described in the Troubleshooting guide, by looking up error messages in the message reference, or looking for related documentation on the online help, contact your local IBM Technical Support in accordance with your actual contract.

WebSphere Application Server comes with a built-in utility that collects logs and configuration information into one file, the Collector Tool. This tool is intended to be used when IBM Technical Support ask you to run it on your environment and submit the output for additional analysis.

11.6.1 First Failure Data Capture tools

The First Failure Data Capture tool preserves the information generated from a processing failure and returns control to the affected engines. The captured data is saved in a log file for use in analyzing the problem.

The First Failure Data Capture tool is intended primarily for use by IBM Service. It runs as part of the IBM WebSphere Application Server, and you cannot start or stop it. It is recommended that you do not attempt to configure the First Failure Data Capture tool. If you experience conditions requiring you to contact IBM Service, your IBM Service representative will assist you in reading and analyzing the First Failure Data Capture log.

The First Failure Data Capture tool does not affect the performance of the IBM WebSphere Application Server.

In the iSeries environment, FFDC logs are located on the directory /QIBM/UserData/WEBAS5/Base/<InstanceName>/logs/ffdc, where <InstanceName> is the name of your instance. Figure 11-32 shows an example of the FFDC log file.

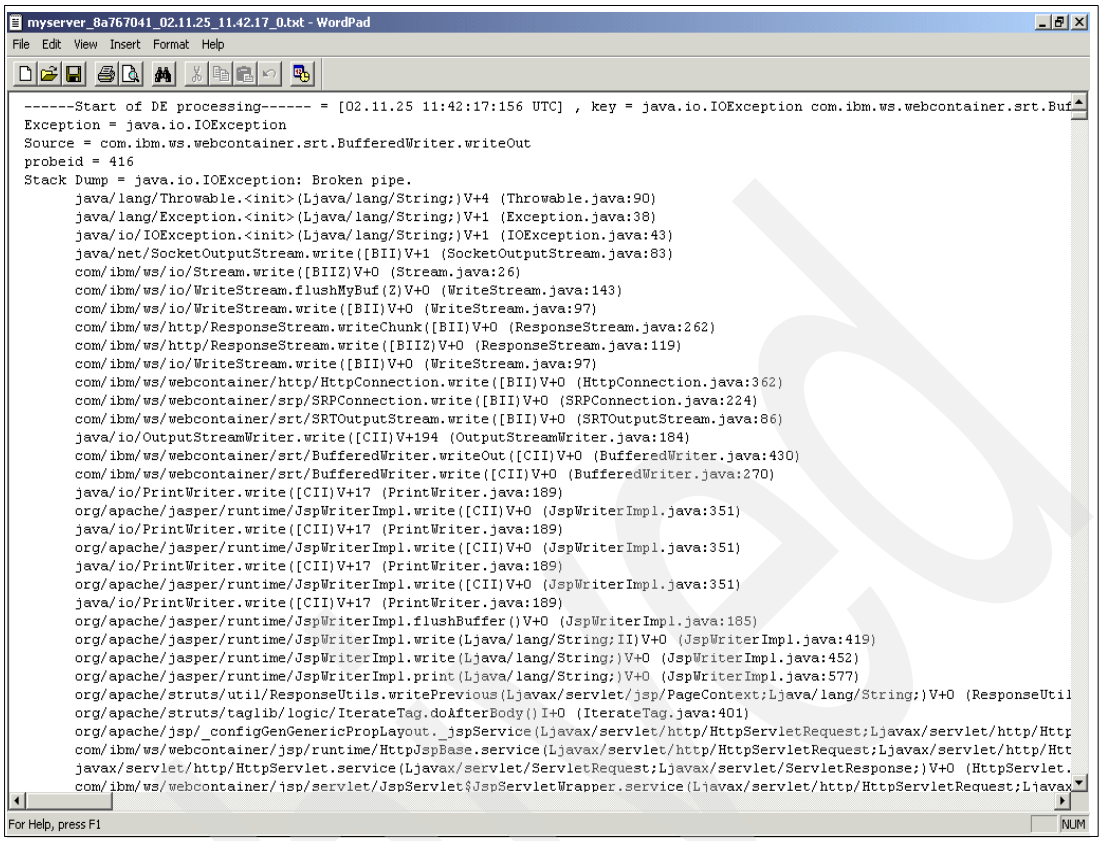


Figure 11-32 FFDC log file example

11.6.2 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. The information includes logs, property files, configuration files, operating system and Java data, and prerequisite software presence and levels.

There are two phases to using the Collector tool. The first phase is to execute the Collector program on your WebSphere Application Server. The second phase is the analysis of the Collector program output .jar file by IBM Customer Support.

The Collector program is designed to run to completion, despite errors such as files or commands not found, in order to collect as much data as possible.

Running the Collector tool

The collector script is available with both WebSphere Application Server and WebSphere Application Server Network Deployment.

Here are the steps for running the Collector tool:

1. Sign on to the iSeries System using a user profile with *ALLOBJ authority.
2. On the iSeries command line, run the Start Qshell Interpreter (STRQSH) command.
3. Change directory to the bin directory for the product: `cd was_install_root/bin`, where `was_install_root` is /QIBM/ProdData/WebAS5/Base for WebSphere Application Server or /QIBM/ProdData/WebAS5/ND for WebSphere Application Server Network Deployment.

4. Run the collector script. The syntax is:
   `collector [-instance instance] [-JarOutName jarfile]`

   The parameters are:

   -instance

   This optional parameter specifies the name of the WebSphere instance for which to collect information. Specify this parameter if you are collecting information for an instance other than the default instance. The default value is default, the default instance.

   -JarOutName

   This optional parameter specifies the fully-qualified path name of the jar file to contain the collector tool output. The location of the jar file must be outside of the /QIBM/ProdData/WebAS5 directory structure for the product. If this parameter is not specified, the default is to create the jar file in the current working directory with the name of the jar file having the following format:

   iSeriesHostName--fullyQualifiedInstancePathName-WASEnv.jar

   If the jar file name exceeds sixty characters, it is truncated to the first sixty characters of the name. It is highly recommended that you specify the -JarOutName parameter.

   Note: The QEJBSVR user profile must have *RWX authority to the directory to contain the output jar file.

Here are some examples:

```
collector -JarOutName /temp/WASEnv.jar
```

Collect information on the `default` WebSphere Application Server instance, placing the output in `WASEnv.jar` located in directory `/temp`.

```
collector -instance myserver -JarOutName /qibm/userdata/webas5/base/myserver/collec.jar
```

Collect information on instance `myserver`, placing the output in collec.jar in directory `/qibm/userdata/webas5/base/myserver`.

Figure 11-33 shows an example of the screen view when you run collector tool.
When you run the collector tool, a collector.log file is created in the directory /QIBM/UserData/WEBAS5/Base/<InstanceName>/logs, where <InstanceName> is the name of instance that you run the collector tool.

```
                                QSH Command Entry

 > collector -instance myserver -JarOutName
/qibm/userdata/webas5/base/myserver/collec.jar
   2002/12/02 13:29:12 URL is:
jar:file:/QIBM/ProdData/WebAS5/Base/lib/collector.jar!/com/ibm/websphere/rastools/collec
tor/Collector
  .class
  2002/12/02 13:29:12 Collector version: 05.00.00
   2002/12/02 13:29:12 Log File name:
/QIBM/UserData/WebAS5/Base/myserver/logs/Collector.log
  2002/12/02 13:29:12 Case sensitive: false
   2002/12/02 13:29:12 Inventory file:
/QIBM/ProdData/WebAS5/Base/properties/default.inventory
  2002/12/02 13:29:23 Compression: 9

 ===>
F3=Exit  F6=Print F9=Retrieve F12=Disconnect
 F13=Clear F17=Top  F18=Bottom  F21=CL command entry
```

Figure 11-33 Example running collector tool

Content Collector Tool output

The collector tool collects the following information about your WebSphere Application Server instance:

- All contents of these directories and any directories below them are gathered in the following directories, and then specific files are collected if a specific file is called out.
  - [instanceRoot]/bin
  - [instanceRoot]/config
  - [instanceRoot]/logs
    Inside this directory it is included the FFDC directory with the FFDC existent logs.
  - [instanceRoot]/properties
  - [instanceRoot]/wstemp
  - /QIBM/ProdData/WebAS5/Base/properties/version/
  - /QIBM/ProdData/WebAS5/Base/classes/properties/systemlaunch.properties
  - /QIBM/ProdData/WebAS5/Base/properties/orb.properties
  - /QIBM/UserData/Java400/SystemDefault.properties
  - /home/QEJBSVR/SystemDefault.properties

  Note: The files, /QIBM/UserData/Java400/SystemDefault.properties and /home/QEJBSVR/System.Default.properties, may or may not exist on your system. If not, the collector tool shows you an error message saying “file does not exist”. This is just an informational message; the collector tool continues to work.

- Output from **DSPPTF** command.
- Output from **DSPSFWRSC** command.
- Output from **DSPHDWRSC *PRC** command.
- Output from **DSPNETA** command.

- Output from **WRKSBS** command.
- Output from **WRKSYSSTS ASTLVL(*ADVANCED)** command.
- Output from **ls -R -la [instanceRoot]**

Results

The Collector program creates a log file, Collector.log, and an output .jar file in the current work directory. The .jar file name is based on the hostname and package of the server on which the Collector tool was run, in the format: hostname-ND|Base-WASenv.jar. See Example 11-1.

Example 11-1 Running the Collector utility

```
> pwd
  /QIBM/ProdData/WebAS5/Base/bin
  $
 > collector
Could not find parameter -instance, using default "default"
2002/12/04 17:42:14 URL is:
jar:file:/QIBM/ProdData/WebAS5/Base/lib/collector.jar!/com/ibm/websphere/rastools/collector
/Collector
.class
2002/12/04 17:42:14 Collector version: 05.00.00
2002/12/04 17:42:14 Log Filename: /QIBM/UserData/WebAS5/Base/default/logs/Collector.log
2002/12/04 17:42:14 Case sensitive: false
2002/12/04 17:42:14 Inventory file: /QIBM/ProdData/WebAS5/Base/properties/default.inventory
2002/12/04 17:42:27 Compression:  9
2002/12/04 17:42:27 User:  null

2002/12/04 17:42:27 EXECUTE: echo
2002/12/04 17:42:28 Execute return code: 0
2002/12/04 17:42:28 Including directory: /QIBM/UserData/WebAS5/Base/default/bin/,
recursively.
2002/12/04 17:42:28 2 objects identified in directory:
/QIBM/UserData/WebAS5/Base/default/bin/
2002/12/04 17:42:28 Including file: /QIBM/UserData/WebAS5/Base/default/bin/setupCmdLine
2002/12/04 17:42:28 Including file: /QIBM/UserData/WebAS5/Base/default/bin/dttest.txt
2002/12/04 17:42:28 Including directory: /QIBM/UserData/WebAS5/Base/default/config/,
recursively.
----------------
 2002/12/04 17:43:28 Including file:
/QIBM/UserData/WebAS5/Base/default/logs/joshwig/SystemOut.log
 2002/12/04 17:43:28 Error - File does not exist:
/QIBM/UserData/WebAS5/Base/default/logs/joshwig/SystemOut.log
 2002/12/04 17:43:28 Including file:
/QIBM/UserData/WebAS5/Base/default/logs/joshwig/SystemErr.log
 2002/12/04 17:43:28 Error - File does not exist:
/QIBM/UserData/WebAS5/Base/default/logs/joshwig/SystemErr.log
 2002/12/04 17:43:28 Including file:
/QIBM/UserData/WebAS5/Base/default/logs/joshwig/native_stdout.log
 2002/12/04 17:43:28 Error - File does not exist:
/QIBM/UserData/WebAS5/Base/default/logs/joshwig/native_stdout.log
 2002/12/04 17:43:28 Including file:
/QIBM/UserData/WebAS5/Base/default/logs/joshwig/native_stderr.log
 2002/12/04 17:43:28 Error - File does not exist:
/QIBM/UserData/WebAS5/Base/default/logs/joshwig/native_stderr.log
 2002/12/04 17:43:28 Including file:
/QIBM/UserData/WebAS5/Base/default/logs/joshwig/trace.log
 2002/12/04 17:43:28 Error - File does not exist:
/QIBM/UserData/WebAS5/Base/default/logs/joshwig/trace.log
```

```
2002/12/04 17:43:28 Including file: /QIBM/UserData/WebAS5/Base/default/logs/activity.log
 2002/12/04 17:43:28 Number possible errors:       8
2002/12/04 17:43:28 Log Filename: /QIBM/UserData/WebAS5/Base/default/logs/Collector.log
 2002/12/04 17:43:28 Output Jar name:
/QIBM/ProdData/WebAS5/Base/bin/RCHAS07.ITSO.IBM.COM--qibm-userdata-webas5-base-default-W
2002/12/04 17:43:28 Including file: /QIBM/UserData/WebAS5/Base/default/logs/Collector.log
 2002/12/04 17:43:29 Return code:  0, Elapsed Time: 00:01:01.746
 $
```

In this example, RCHAS07.ITSO.IBM.COM--qibm-userdata-webas5-base-default-W is the name of the jar file. You can view the jar file by using a file decompressor utility, such as WINZIP. Sample output of the JAR is shown in Figure 11-34.

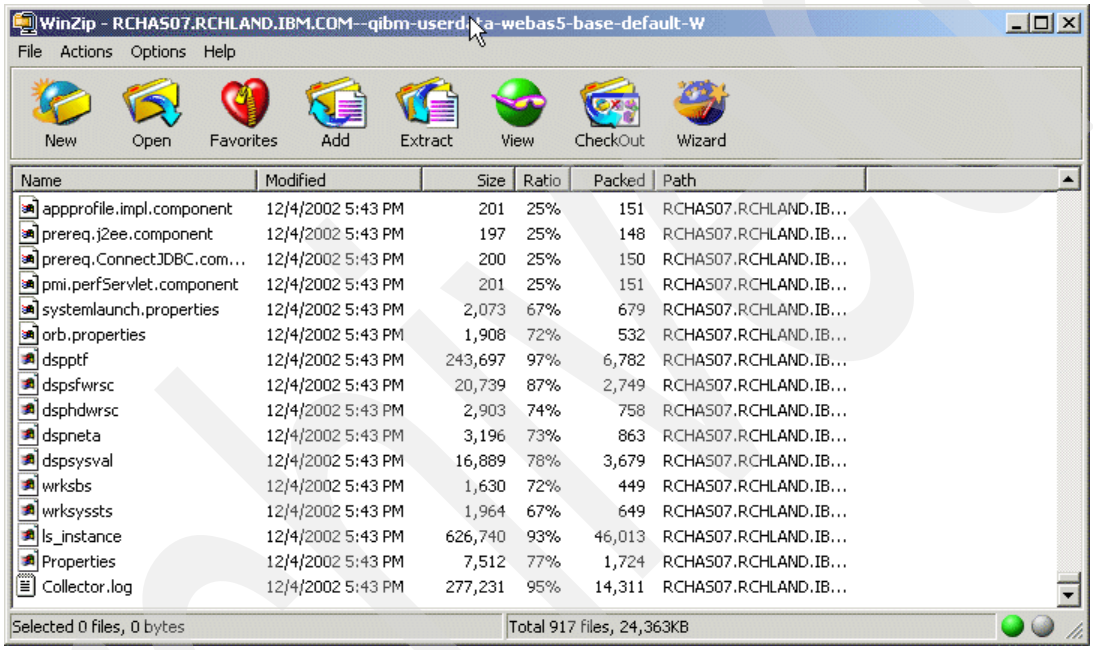


Figure 11-34 Sample content of collector output

11.7 Performance Trace Data Visualizer

The IBM Performance Trace Data Visualizer (PTDV) is a Java application that can be used for performance analysis of applications running on iSeries. PTDV works with the Performance Explorer (PEX) component of the iSeries base operating system, which is used to make performance data collection of our iSeries applications and provides profiling, statistical and trace information.

Performance Trace Data Visualizer (PTDV) allows the analyst to view program flows and get details such as CPU time, wall clock time, number of cycles, and number of instructions, summarized by trace, job, thread, and procedures. When visualizing Java application traces, additional details such as the number and type of objects created, and information about Java locking behavior can be displayed. There is also support for PEX events generated by the WebSphere Application Server. PTDV allows sorting of columns, exporting of data, and many levels of data summarization.

In the following sections we provide an example of how to use this tool in your environment, when you need to know, at a deep level, how your application is running.

11.7.1 Modes of Operation of Performance Trace Data Visualizer

There are four basic modes of operation for PTDV.

Thin Client

This is the default mode for PTDV. In this mode, there is a PTDV server which runs on your iSeries, and does most of the data processing. Then the PTDV client acts mostly as a presentation layer on your client system. This mode is generally the fastest method of running the PTDV, as all of the database access is done on the local system. Using this mode also allows you to process the largest number of events (currently the largest trace processed has been about 2.5 million events), as the data is kept on the iSeries, which can generally handle large amounts of data more effectively than a PC.

The primary disadvantage to this mode is that a connection to your iSeries must be active the entire time you are using PTDV, and some processing time on the server will be used. Therefore, this method is not ideal for viewing data on production systems. In addition, the PTDV server must be installed on the iSeries, which is sometimes not practical/desirable.

If you choose this client, there must be two additional jobs in the iSeries system, running in the QBATCH subsystem, called PTDVSERVER and QJVACMDSVR. Figure 11-35 shows an example of this jobs. These jobs will be automatically started when you select Thin client in PTDV client workstation.

```
                          Work with Active Jobs                  AS07
                                                              12/09/02  16:41:33
CPU %:    8.9     Elapsed time:   03:02:58     Active jobs:   291
Opt  Subsystem/Job  User        Type  CPU %  Function        Status
     QBATCH         QSYS        SBS      .0                 DEQW
       PTDVSERVER   ITSCID16    BCH      .0  CMD-JAVA        TIMW
       QJVACMDSRV   ITSCID16    BCI      .0  PGM-QJVACMDSRV  JVAW
     QCMN           QSYS        SBS      .0                 DEQW
     QCTL           QSYS        SBS      .0                 DEQW
       QSYSSCD      QPGMR       BCH      .0  PGM-QEZSCNEP    EVTW



                                                                More...
===>
F21=Display instructions/keys
```

Figure 11-35 Jobs for PTDV Thin client

Thick Client

This mode is very similar to the way that the original version of PTDV ran. In this case, the data processing code and the presentation are both done on your client system, and the iSeries is used only to access the database. This mode tends to be somewhat slower than thin-client mode, and can require more network bandwidth. In addition, the amount of data which can be processed is limited to the amount of memory on your PC — most systems will run out of memory if more than 300,000 events (or sometimes even less) are processed.

However, the thick client mode can be useful when dealing with small collections (of only a few thousand events), since it does not require any PTDV code to be installed on the iSeries. Since it does not require a connection to the server after the events are initially processed, this mode can also be useful when it is not acceptable to do processing on the server over an extended period of time.

Three tier

This mode is a bit of a mix between thin client and thick client. In this case, one iSeries holds the PEX data. Another iSeries runs the PTDV server, and your client system runs the PTDV client. This results in a mix of the advantages and disadvantages above. Processing can sometimes take even longer than in the Thick Client mode. However, the Three Tier mode can process as many events as the Thin Client mode.

In addition, the connection to the first iSeries (database server) is closed after the initial processing (as in Thick Client), so this mode can work well for retrieving data from production systems. In general, the Thin Client mode is preferable unless it is necessary to do only minimal processing on the system with the database. Even in this case, it is generally better to end PEX with the *FILE option, and then move the database over to another iSeries system which can be used for the processing.

Manual

This mode is generally only used for debugging purposes. All of the properties that are used in accessing the PTDV data are set manually. Thus, this mode can be used to emulate any of the other modes (although emulating Thick Client will not perform as well, because we are normally able to make some special optimizations for Thick Client). Using Manual Mode could allow extra flags to be added to the database URL, or the use of non-standard iSeries JDBC (Java DataBase Connectivity) drivers. So far, we have never needed this mode even for debugging, but it is conceivable that we might need it some day, so we have left it in. No further documentation is available for Manual Mode — if you really want to try it, you're on your own.

11.7.2 Installing Performance Trace Data Visualizer (PTDV) in client server

If you want to use this tool as part of the troubleshooting process or to identify the performance of your application, you must do the following steps.

1. Download PTDV zip file. Performance Trace Data Visualizer zip file must be download from:

   `http://alphaworks.ibm.com/tech/ptdv`

2. Install a Java2 runtime on your client system. The latest IBM Java runtimes for a variety of platforms can be downloaded from:

   `http://www.ibm.com/developerworks/java/jdk/`

   You can verify that you have installed the JVM properly by opening a DOS windows (or shell prompt) and running the following command:

   ```
   java - version
   ```

   Note: If you installed the WebSphere workstation tools in the client system, you can use the JVM included on these tools.

3. Install the IBM Toolbox for Java (or JTOpen). You can download it from:

   `http://www.ibm.com/servers/eservers/iseries/toolbox/download.htm`

   You must copy the `jt400.jar` files to your extensions directory. The location of your extensions directory depends on the exact Java Runtime Environment you have installed, and the directory you installed it to. For example, the default installation directory for Workstation tools's JDK on Windows is:

   ```
   <drive>:\program files\WebSphere\Appserver\java\jre\lib\ext
   ```

   Here, `<drive>` is the name of the local drive in the client system.

4. Unzip the ptdv.zip file. The installation process asks you for the directory where Java is installed, as shown in Figure 11-36.

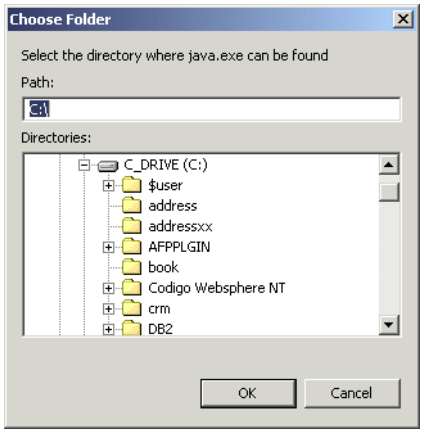


Figure 11-36 Installing PTDV in client system

5. The installation for PTDV is started as shown in Figure 11-37.

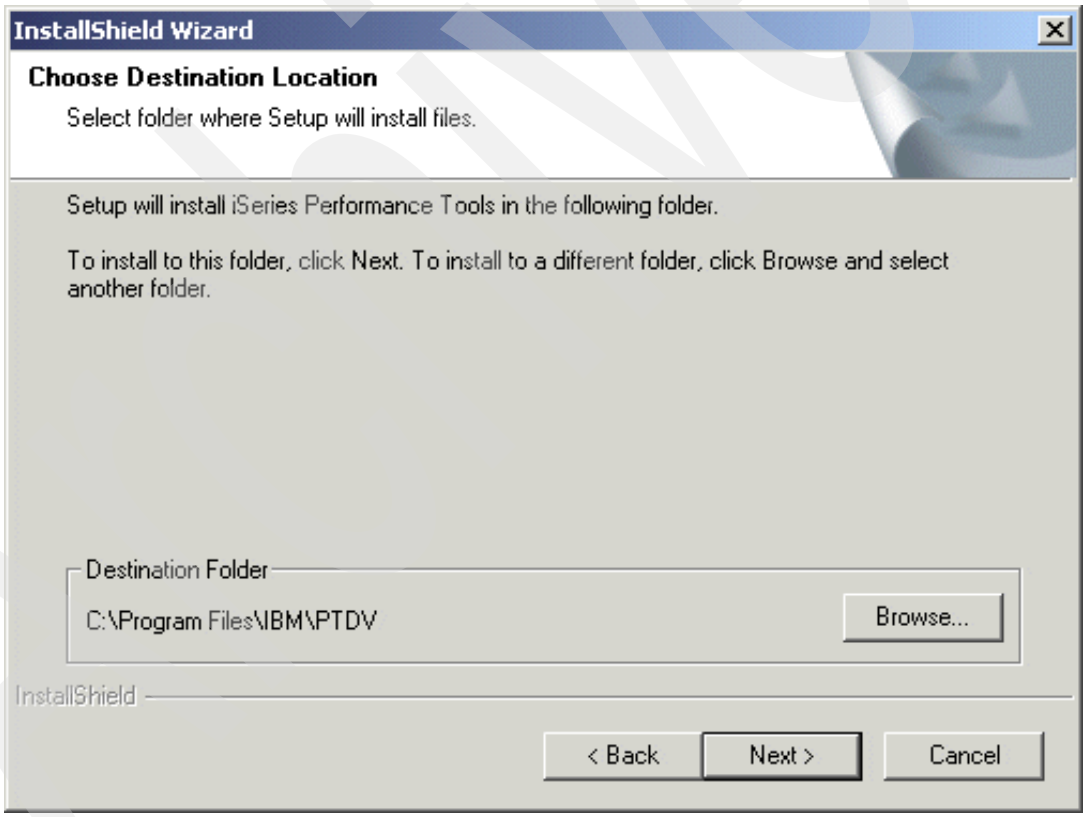


Figure 11-37 PTDV installation

6. The installation process unzips all the files required by PTDV and shows a window notifying you of the completion of the installation process.
7. Click **Finish**.

11.7.3 Requirements for Performance Trace Data Visualizer on iSeries system

The requirements for this vary, depending on which mode(s) you will be running PTDV in:

- Thin Client:
  - OS/400 V5R1 or later
  - TC1 - TCP/IP Connectivity Utilities
  - JV1 - Developer Kit for Java
- Thick Client:
  - OS/400 V5R1 or later
  - TC1 - TCP/IP Connectivity Utilities
  - Host Servers option of OS/400
- Three-Tier:

  When running Three-Tier, there are two iSeries systems involved: the database server, and the system which will run the PTDV server.

  - Database Server:
    - OS/400 V5R1 or later
    - TC1 - TCP/IP Connectivity Utilities
    - Host Servers option of OS/400
  - PTDV Server:
    - OS/400 V5R1 or later
    - TC1 - TCP/IP Connectivity Utilities
    - JV1 - Developer Kit for Java
    - JC1 - Toolbox for Java

11.7.4 Installing Performance Trace Data Visualizer on the iSeries system

If you run PTDV in Thin Client, Three-Tier, or Manual mode, you need to install the PTDV Server on your iSeries. You must the following steps:

1. Download PTDV Server (`ptdvlib.zip` file) from:

   `http://alphaworks.ibm.com/tech/ptdv`

2. Unzip the `ptdvlib.zip` file in your client system. This process should unzip two files: ptdv5r1.savf and ptdv5r2.zip. Depending upon the operating system installed in your iSeries, you must choose between these two savefiles. In the following steps we work with the ptdv5r2.savf file.

3. On the iSeries system, run the following commands:

   a. `CRTLIB PTDVDIST`

   b. `CRTSAVF PTDVDIST/PTDV5R2`

   c. Transfer ptdv5r2.savf to your iSeries system via FTP. From your client system (from the directory which you unzipped ptdv.zip file), enter the following commands:

      i. `ftp myiSeries`

      Here, myiSeries is the name of your iSeries system. You can use also the iSeries ip address.

      ii.Enter username/password

      iii.cd ptdvdist

      iv.binary

      v. put ptdv5r2.saf ptdv5r2.ptdv5r2 (replace

      vi.quit

4. `CRTLIB PTDV`
5. `RSTOBJ OBJ(*ALL) SAVLIB(PTDV) DEV(*SAVF) SAVF(PTDVDIST/PTDV5R2)`
6. `MKDIR DIR('/QIBM/ProdData/iDoctor/PTDV')`
7. Copy the ptdv.jar file from the client system to the /QIBM/ProdData/iDoctor/PTDV directory by running the following commands:
   a. `ftp myiSeries`

      Here, myiSeries is the name of your iSeries system. You can use also the iSeries ip address.
   b. `Enter username/password`
   c. `cd` QIBM/ProdData/iDoctor/PTDV
   d. `binary`
   e. `put ptdv.jar`
   f. `quit`

11.7.5 PEX definitions

The PEX tool allows you to define different kinds of data to be collected. Next we show you the main topics that you must consider when want to collect information about your application.

Using PTDV with Java

PTDV can see Java method information for methods running with the interpreter, the Just In Time (JIT) compiler, or Direct Execution (DE). See the iSeries Java documentation for information on selecting which mode to use when running a Java program.

In order to see Java method information in the Trace Visualizer, entry/exit hooks must be enabled in the Java classes. There are two basic methods of doing this:

- Run your program with the JIT (or interpreter), and enabled method entry/exit hooks. One way to force your program to run with the JIT is to set the property java.compiler=jitc. To enable performance hooks, you can set os400.enbpfrcol=1.
- Enable method entry/exit hooks in transformed (aka DE'd) Java code. In order to do this, you will need to use CRTJVAPGM with the ENBPFRCOL(*ENTRYEXIT) option to create Java program objects with hooks. You will need to do this for each jar, zip, or class file you want hooks for. The Trace Visualizer can view method information for DEd code at any optimization level, but level 40 is preferred, since this is what your application should normally be running at. Using CRTJVAPGM on large files or on slow machines may result in a long wait. If you are running on a machine with a greater batch than interactive capacity, be sure to submit the CRTJVMPGM to batch.

  Note that only classes loaded via the system or bootstrap class loaders can run under Direct Execution mode. Classes loaded with a user class loader (such as servlets running with WebSphere) will always run with the JIT compiler (unless you explicitly place these classes in the system classpath).

If you are interested in seeing what object types are being created and/or which object types are being locked, you can use a PEX definition containing the *OBJCRT and *LCKSTR Java events. Object information can be collected with or without enabling entry/exit hooks. For more information on setting up PEX definitions, see Performance Explorer Trace event at:

`http://www.ibm.com/servers/eserver/iseries/perfmgmt/resource.htm`

Restriction: Do not attempt to run the CRTJVAPGM command against the java.zip, sun.zip, classes.zip, or rt.jar files supplied by IBM on your iSeries. If you do, they will become corrupted and you will need to re-install the JV1 product to recover. You can run CRTJVAPGM against the classes used by the native JDBC driver (part of JV1), the Java Toolbox classes installed with JC1, and the open-source version of the Java Toolbox (JTOpen). Each release of IBM's WebSphere Application Server contains several jar files that need special options if they are to be re-created with hooks using CRTJVAPGM. Therefore, you should not use CRTJVAPGM on jar files shipped with WebSphere. In all of these cases, you can force use of the JIT compiler and enable hooks using the first method listed above to see performance information on these classes.

Using PTDV with ILE Languages

If you want to trace ILE languages, including programs called from Java using JNI (Java Native Interface), you can use a PEX definition including the *MIENTRY, *MIEXIT, *MISTR, and *MIEND for PGMEVT events. You will then need to add entry/exit hooks in those programs using the CHGPGM command and specifying the ENBPFRCOL(*ENTRYEXIT) option.

Using PTDV to log WebSphere events

WebSphere Application Server for iSeries can generate PEX events which can be processed with PTDV. WebSphere contains a performance data gathering mechanism called PM. The generation of PEX events is integrated with this support on iSeries. This has the advantage of being able to dynamically enable and disable events for different WebSphere components at a detailed level, but with the disadvantage of a slightly increased level of overhead.

WebSphere Application Server V5 is actually capable of generating two types of PEX trace events: WebSphere Trace events (Defining APPEVT *WAS in your PEX definition), and Transaction events (*usrtns). Use of the WebSphere Trace events is recommended.

Creating a PEX definition

The library PTDV contains a CLP source program (PTDVDFN5R2 on the source file QCLSRC) which you can compile in order to generate the standard PEX definition for the most common trace data requirements. You can compile this CLP program and then run it to generate the PEX definition. Figure 11-38 shows an example of a PEX definition tracing all Java events related to WebSphere Application Server and the user transactions.

```
ADDPEXDFN DFN(PTDVALL) TYPE(*TRACE) JOB(*ALL) MAXSTG(1000000) +
   TRCTYPE(*SLTEVT) SLTEVT(*YES) +
   MCHINST(*SIGJXEXP) +
   BASEVT(*PMCO) +
   PGMEVT(*JVAENTRY *JVAEXIT *MISTR *MIEND) +
   JVAEVT(*OBJCRT *LCKSTR *UNLCK *THDCRT *THDDLT *CLSLOAD *GBGCOLSWEEP) +
   APPEVT(*WAS) +
   TEXT('PTDV All Java events')

ADDPEXDFN DFN(PTDVJAVA) TYPE(*TRACE) JOB(*ALL) MAXSTG(1000000) +
   TRCTYPE(*SLTEVT) SLTEVT(*YES) +
   BASEVT(*PMCO) +
   PGMEVT(*JVAENTRY *JVAEXIT) +
   JVAEVT(*OBJCRT *LCKSTR *UNLCK) +
   APPEVT(*WAS) +
   TEXT('PTDV Java methods, obj creates, locks, WAS')
```

Figure 11-38 PEX definition example to trace WebSphere Application Server events

There are some recommendations about the PEX definition in order to capture the right data that you would need to analyze using the PTDV tool:

- It is recommended that you limit your trace to 1,000,000 events or less, though more than that can be collected and processed successfully by the PTDV. When running PTDV in Thick Mode, the recommended limit is 200,000 events. You can create a PEX definition including only the events you need.
- If multiple jobs are running on the system, change the PEX definition to collect events only for the job you are interested in. You can use the `CHGPEXDFN` command to do this, and specify the job name in the `JOB` parameter.
- If you do need method entry/exit events in your trace, try to add entry/exit hooks only to the classes that you need to see the events for. For example, if you are using the CRTJVAPGM ENBPFRCOL(*ENTRYEXIT) method of generating hooks, only generate them for the jar files or classes that you are interested in. If the part of your code you are interested in normally runs in the JIT, then you can enable hooks for the JIT, but don't force all of the other code to run in the JIT. (In other words, specify the os400.enbpfrcol=1 property, but not java.compiler=jitc.) This can be particularly helpful when analyzing code which runs under WebSphere, because you can enable hooks for your application code (which normally runs with the JIT) but not for the WebSphere code (which normally runs DE).
- Run only a small number of transactions or invocations during the trace. For example, if you are analyzing a Java servlet, try just loading the servlet a couple times manually from your Web browser, rather than tracing under a full load.

  This method will not be effective for analyzing Java lock problems, since the locking behavior of your application may change significantly under heavier loads. But it can work very well for path length or object creation analysis.
- If none of the other methods will work, sometimes the best thing to do is just to start and end PEX as quickly as possible. One method of doing this is to start your program and let it ramp up to steady state. Then type the STRPEX and ENDPEX statements into two separate windows so that you can do the STRPEX, quickly switch windows, and simply press Enter to do the ENDPEX.

11.7.6 Performance Monitoring Infrastructure

The Performance Monitoring Infrastructure (PMI) uses a client-server architecture. The server collects performance data from various WebSphere Application Server components. A client retrieves performance data from one or more servers and processes the data.

As shown in Figure 11-39, the server collects PMI data in memory. This data consists of counters such as servlet response time and data connection pool usage. The data points are then retrieved using a Web client, Java client or JMX client. WebSphere Application Server contains Tivoli Performance Viewer, a Java client which displays and monitors performance data. The right side of this figure displays the server updates and keeps PMI data in memory. The left side displays a Web client, Java client and JMX client retrieving the performance data.

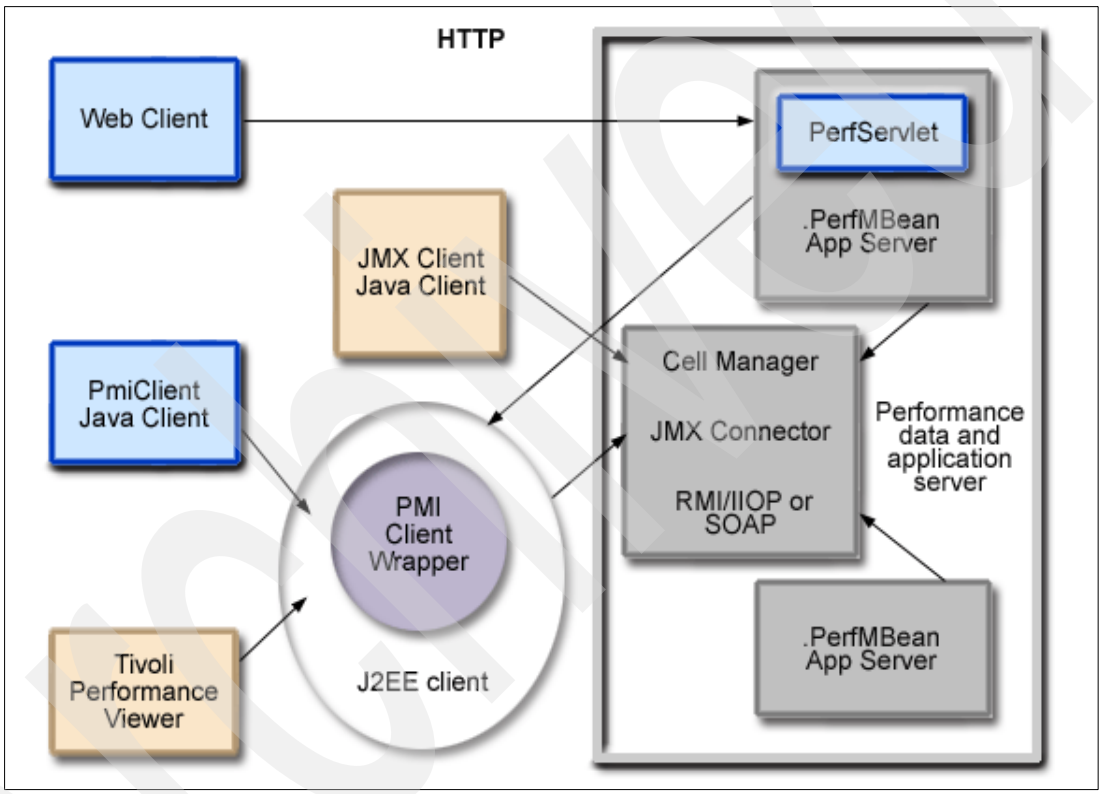


Figure 11-39 PMI architecture

Enabling Performance Monitor Services in WebSphere Application Server

You must enable Performance Monitor Services (PMS) in your application server instance, in order to capture WebSphere trace information in your PEX definition. You must do the following steps in order to enable PMS.

1. Start the administrative console. For more information, see Start the administrative console.
2. Expand **Servers** in the left column.
3. Click **Application Servers** in the left column.
4. Click on the server for which you want to enable performance monitoring.
5. Click the **Configuration** tab.
6. Click **Performance Monitoring Service**.

7. Select **Startup.**
8. Select the desired Initial specification level. Options are None, Standard or Custom.
9. Click **Apply** or **OK**.
10. Click **Save** to save the configuration.
11. Restart the application server. The changes made do not take effect until you restart the application server.

Figure 11-40 shows an example PMS configuration using the administrative server console.

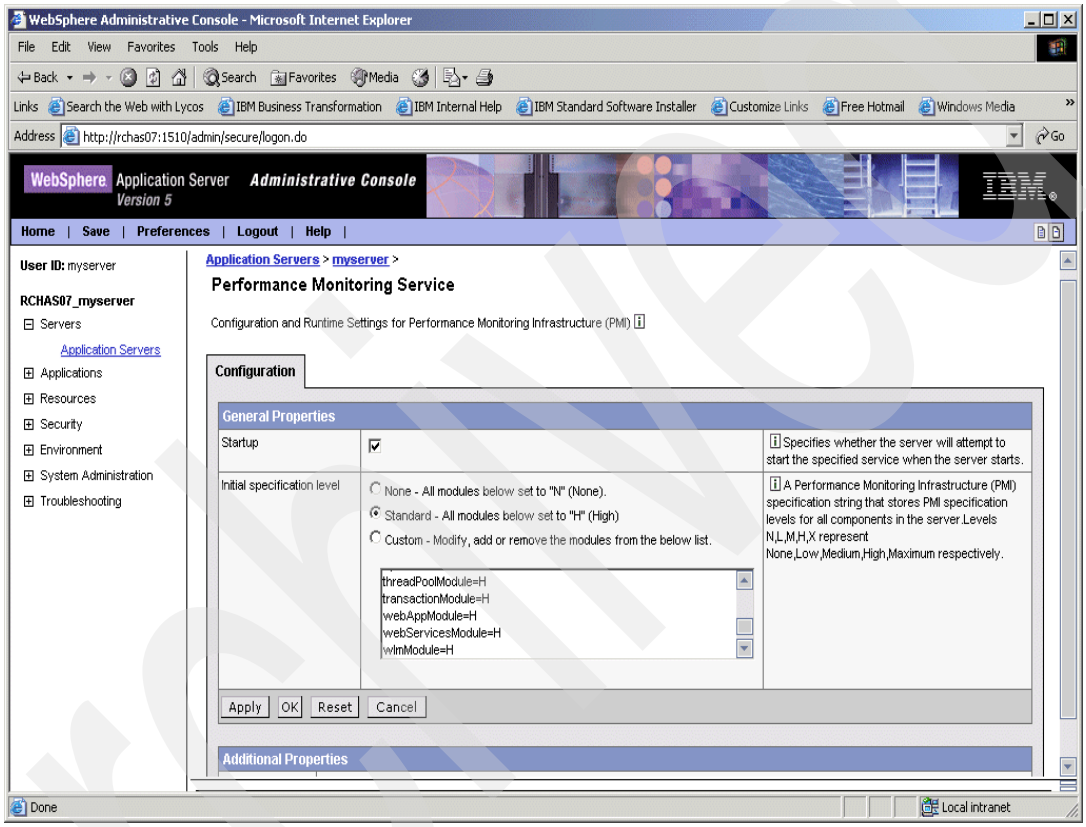


Figure 11-40 Enabling PMI services

You can choose the specification level for each server components. You can select None, standard or custom if you want a different trace level for some specific components. For custom choice you must specify any of the following levels:

- Null=N
- Low=L
- Medium=M
- High=H

The server components available are:

- beanModule
- cacheModule
- connectionPoolModule
- j2cModule
- jvmRuntimeModule
- orbPerfModule
- servletSessionsModule

- systemModule
- threadPoolModule
- transactionModule
- webAppModule
- webServicesModule
- wlmModule
- wsgwModule

11.7.7 Using PTDV to analyze a trace

As we saw in the previous topic, PTDV has four different kinds of clients that you can choose, depending on your needs. Refer to 11.7.1, “Modes of Operation of Performance Trace Data Visualizer” on page 483 for more information. You must execute the following steps to analyze your PEX collection using PTDV.

Note: When you start the PTDV tool in the client system, you can see a DOS window, where all messages related to PTDV are logged. This window helps you to find out what has happened with the PTDV analysis of the PEX collection.

1. Start PTDV in the client workstation.
2. PTDV displays a window where you can choose what kind of client you want to start. Figure 11-41 shows an example of this window.

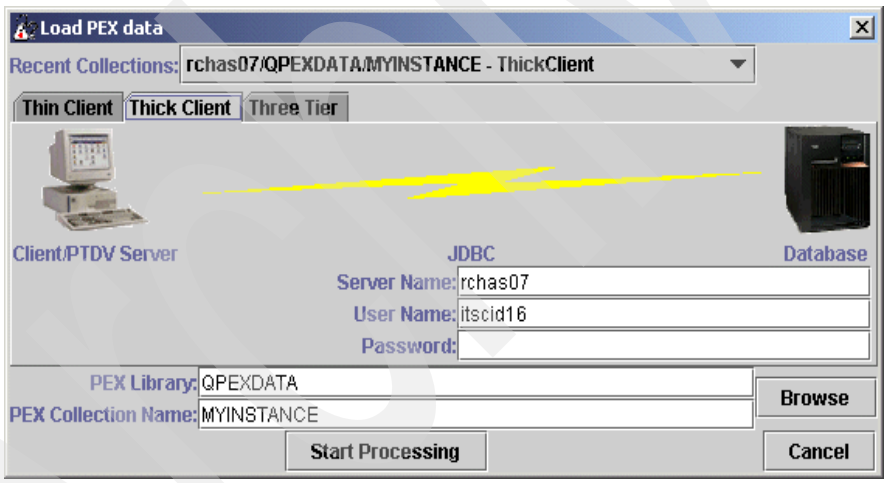


Figure 11-41 Choosing the PTDV client

This window asks you to fill in the information about server name, user ID, and password for the iSeries system. Also, you must select the PEX library and PEX collection to analyze. You can use the **Browse** button in order to do the PTDV connection to the iSeries system and see a list of all PEX collections available in the library chosen.

3. PTDV starts to load the information. This could take some time, depending on the number of events in your PEX collection.

4. Next, PTDV shows you five windows containing information related to the PEX collection:
   - **Cumulative information:** This contains summary information about your iSeries system and your PEX collection. You can see an example of this in Figure 11-42.

     Notice that there are two options available from this window:
     - Export Table. Allow you to save the contents of this window on a text file.
     - Copy to clipboard. Allow you to copy the contents of this window into the windows clipboard.

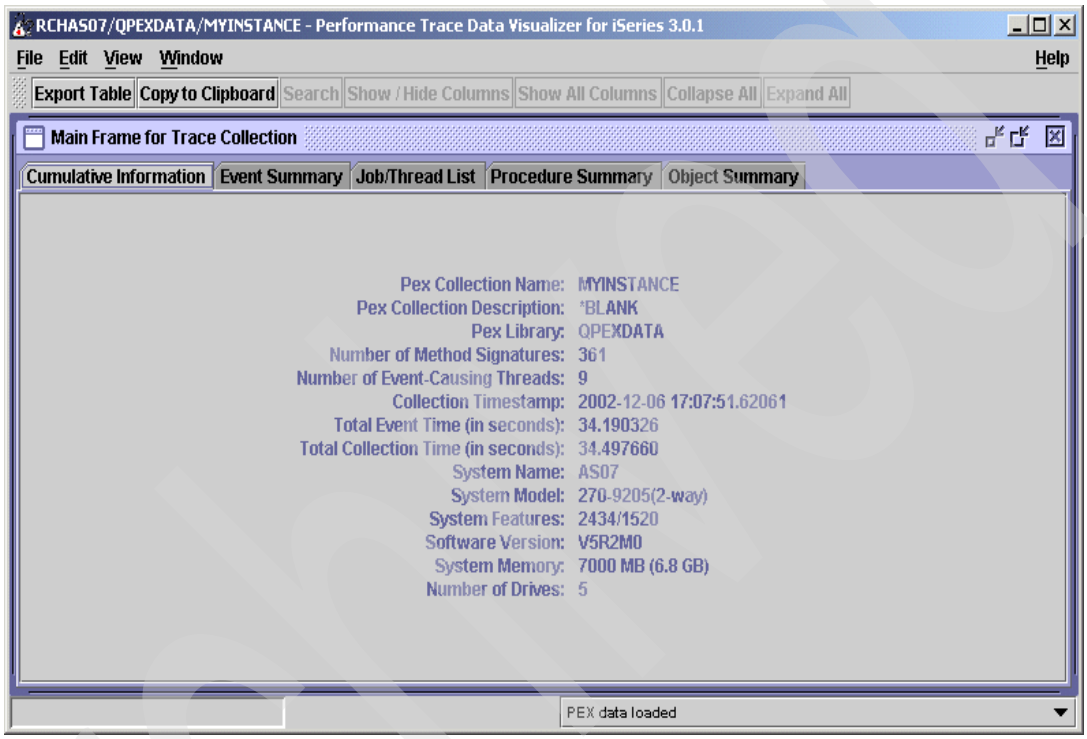


Figure 11-42 PTDV cumulative Information window

   - **Event summary:** This shows you a summary of all events included in the PEX collection. It shows the total events classified by event type, and the quantity of events processed by PTDV tool. Figure 11-43 shows an example of this window. There are seven additional options that you can use in this window:
     - Export
     - Copy to Clipboard
     - Search
     - Show/Hide Columns
     - Show All Columns
     - Collapse All
     - Expand All

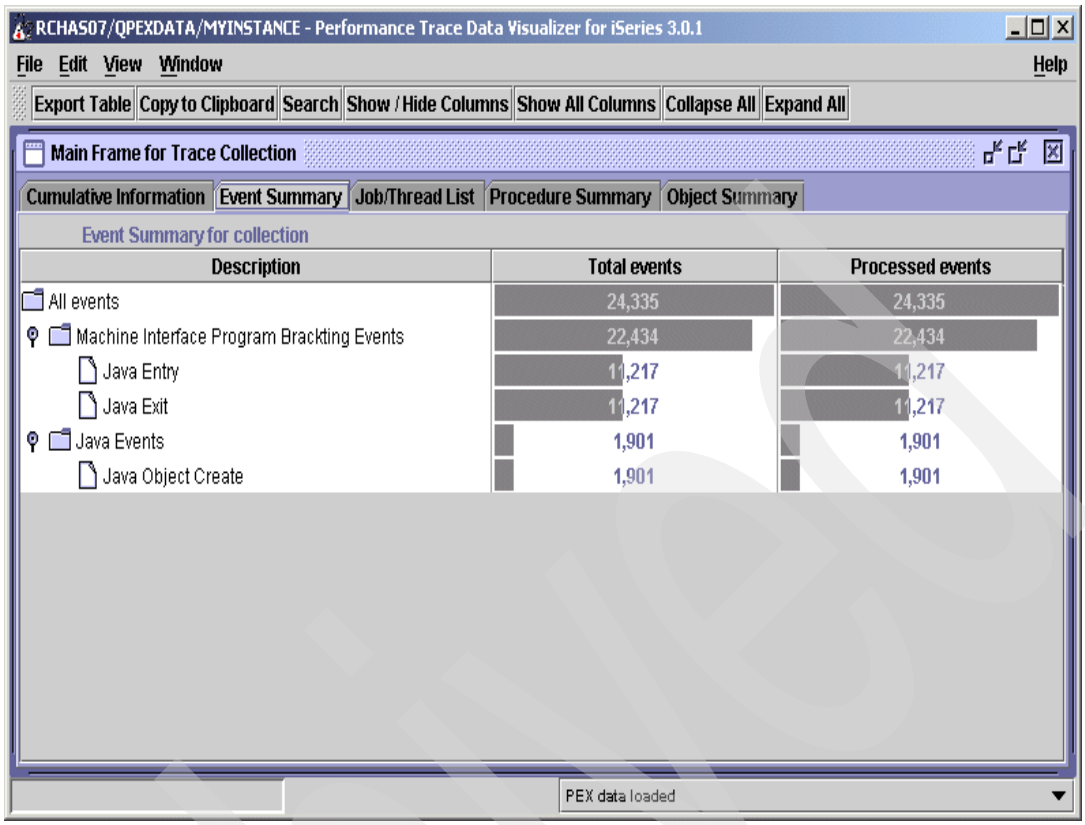


Figure 11-43 PTDV Event Summary window

– **Job/thread list:** This is a list of all jobs and threads included in the PEX collection. Figure 11-44 provides an example of this window. It shows the total events classified by event type, and the quantity of event processed by PTDV tool, cumulative CPU time (nanoseconds), active time (nanoseconds), and cumulative cycles. There are seven additional options that you can use in this window:

  - Export
  - Copy to Clipboard
  - Search
  - Show/Hide Columns
  - Show All Columns
  - Collapse All
  - Expand All

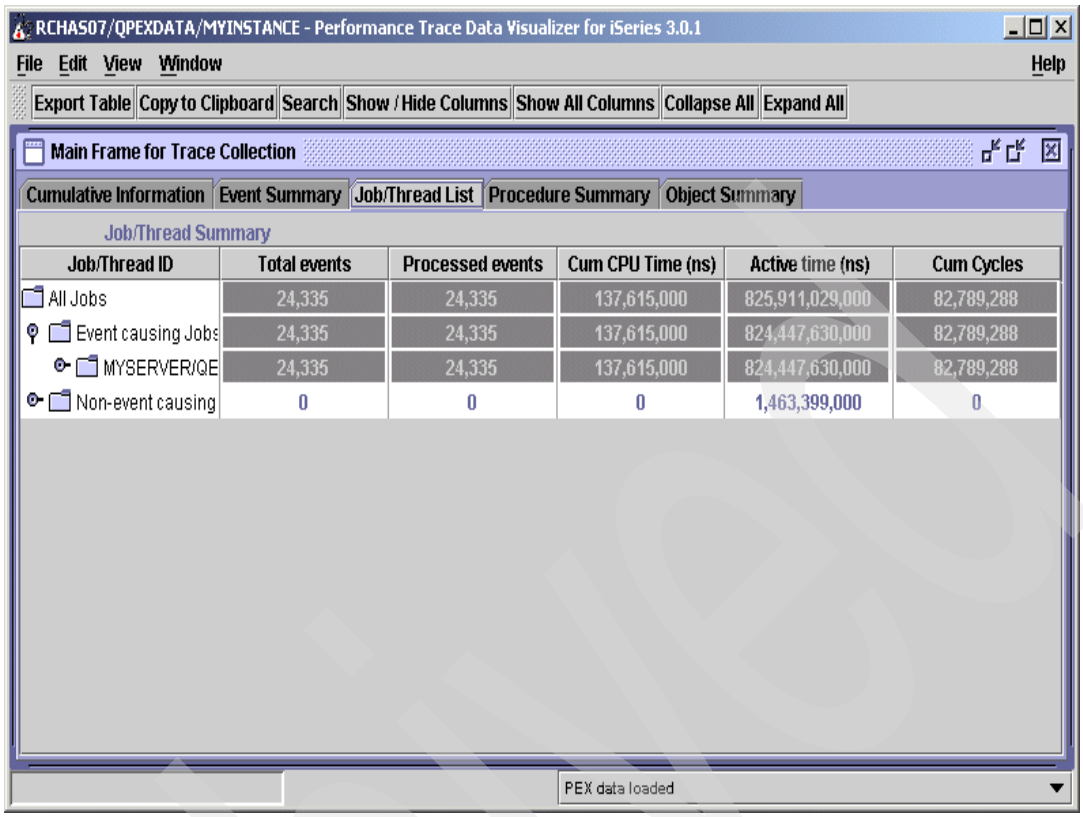


Figure 11-44 PTDV Job/Thread List window

– **Procedure summary:** This shows you a list of all procedures called during the PEX trace. Figure 11-45 provides an example of this window. You can use the **Show/Hide Columns** option to see more detailed information related to the procedures.

  - Name
  - Invocations
  - Inline Time (ns)
  - Avg Inline Time (ns)
  - Cum Time (ns)
  - Inline task execution cycles
  - Cum task execution cycles
  - Avg Inline Cycles
  - Inline Instructions
  - Cum Instructions
  - Inline CPI
  - Inline Creates
  - Cum Creates
  - Inline lock events
  - Cum lock events
  - Inline Deletes
  - Cum Deletes

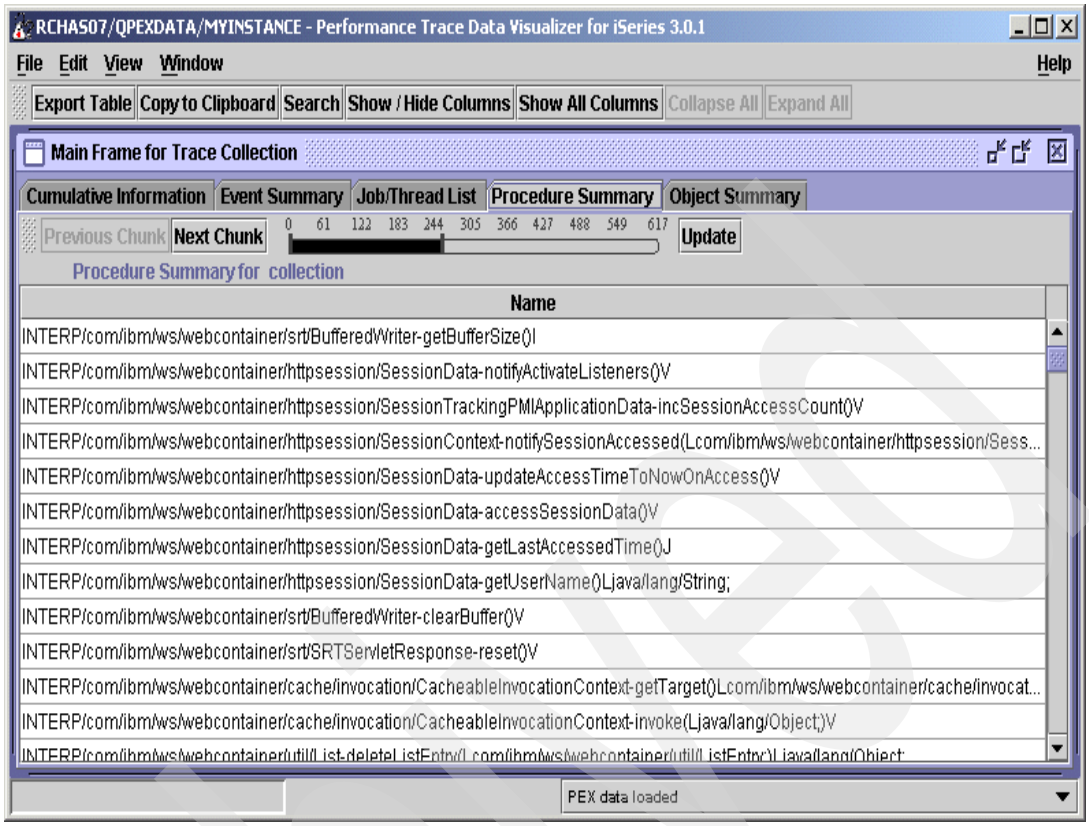
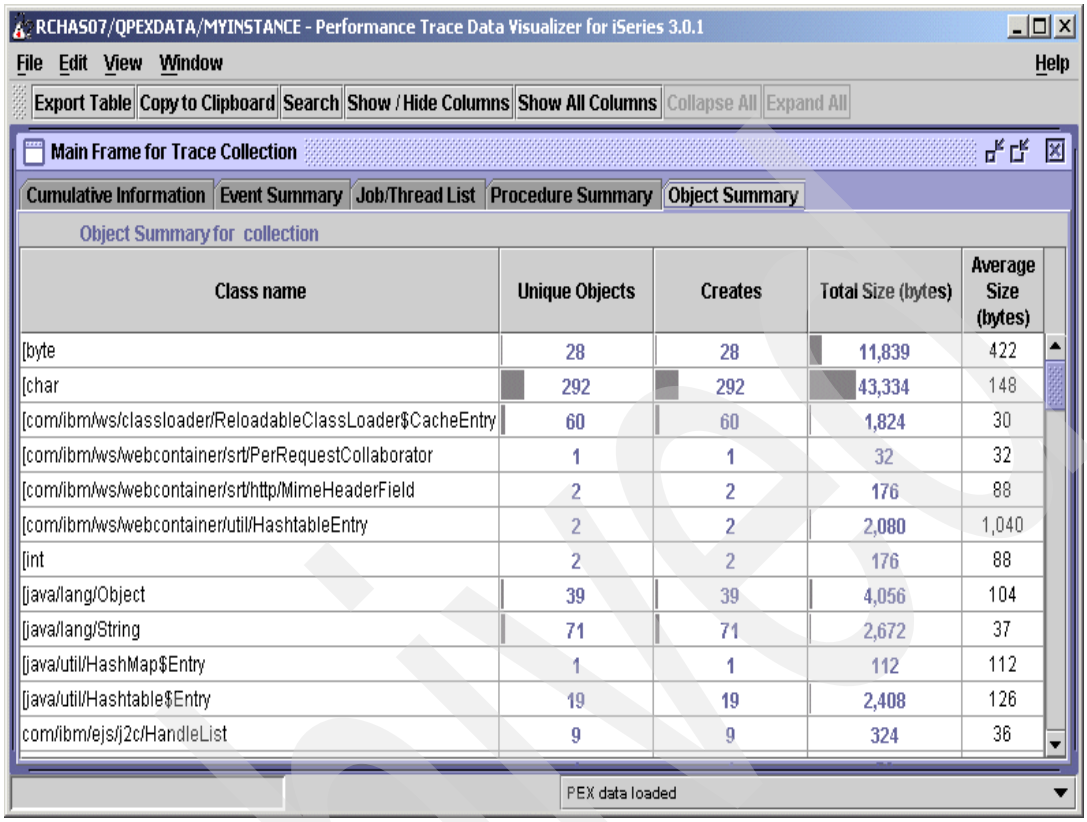


Figure 11-45 PTDV Procedure summary

– **Object summary:** This shows you a list with all objects used during the PEX trace. Figure 11-46 shows an example of this window. You can use the **Show/Hide Columns** option to see more detailed information related to the objects:

- Class name
- Unique Objects
- Creates
- Total Size (bytes)
- Average Size (bytes)
- Class loader name
- Locks
- Lock events
- Min Size (bytes)
- Max Size (bytes)
- Deletes
- Created but not Deleted
- Deleted but not Created
- Created and Deleted
- Creates - Deletes
- Total hold time (ns)
- Total wait time (ns)
- Max hold time (ns)
- Max wait time (ns)
- Min hold time (ns)
- Min wait time (ns)
- Avg hold time (ns)
- Avg wait time (ns)

- Thread Notify/Wait events
- Opt level



Class name	Unique Objects	Creates	Total Size (bytes)	Average Size (bytes)
[byte	28	28	11,839	422
[char	292	292	43,334	148
[com/ibm/ws/classloader/ReloadableClassLoader$CacheEntry	60	60	1,824	30
[com/ibm/ws/webcontainer/srt/PerRequestCollaborator	1	1	32	32
[com/ibm/ws/webcontainer/srt/http/MimeHeaderField	2	2	176	88
[com/ibm/ws/webcontainer/util/HashtableEntry	2	2	2,080	1,040
[int	2	2	176	88
[java/lang/Object	39	39	4,056	104
[java/lang/String	71	71	2,672	37
[java/util/HashMap$Entry	1	1	112	112
[java/util/Hashtable$Entry	19	19	2,408	126
com/ibm/ejs/j2c/HandleList	9	9	324	36

Figure 11-46 PTDV Object summary

11.7.8 Example using the PTDV tools

Here we show an example of how to use the PTDV tools. Basically, this collects all of the information shown in the previous topics.

1. Prepare the WebSphere Application Server for Performance collection:
   a. You must add a system property for the Java Virtual Machine on your WebSphere Application Server. Refer to Figure 11-47 for an example of the administrative console window:
      i. Start the administrative console. For more information, see “Start the administrative console”.
      ii. Expand **Servers** in the left column.
      iii. Click **Application Servers** in the left column.
      iv. Click the server for which you want to enable performance monitoring.
      v. Click the **Configuration** tab.
      vi. Click **Process Definition**.
      vii. Click **Java Virtual Machine**.
      viii. Click **Custom** properties.
      ix. Click **New**.
      x. Write `os400.enbpfrcol` in name field and `1` in value field.

xi. Click **Apply** or **OK**.

xii. Click **Save** to save the configuration.

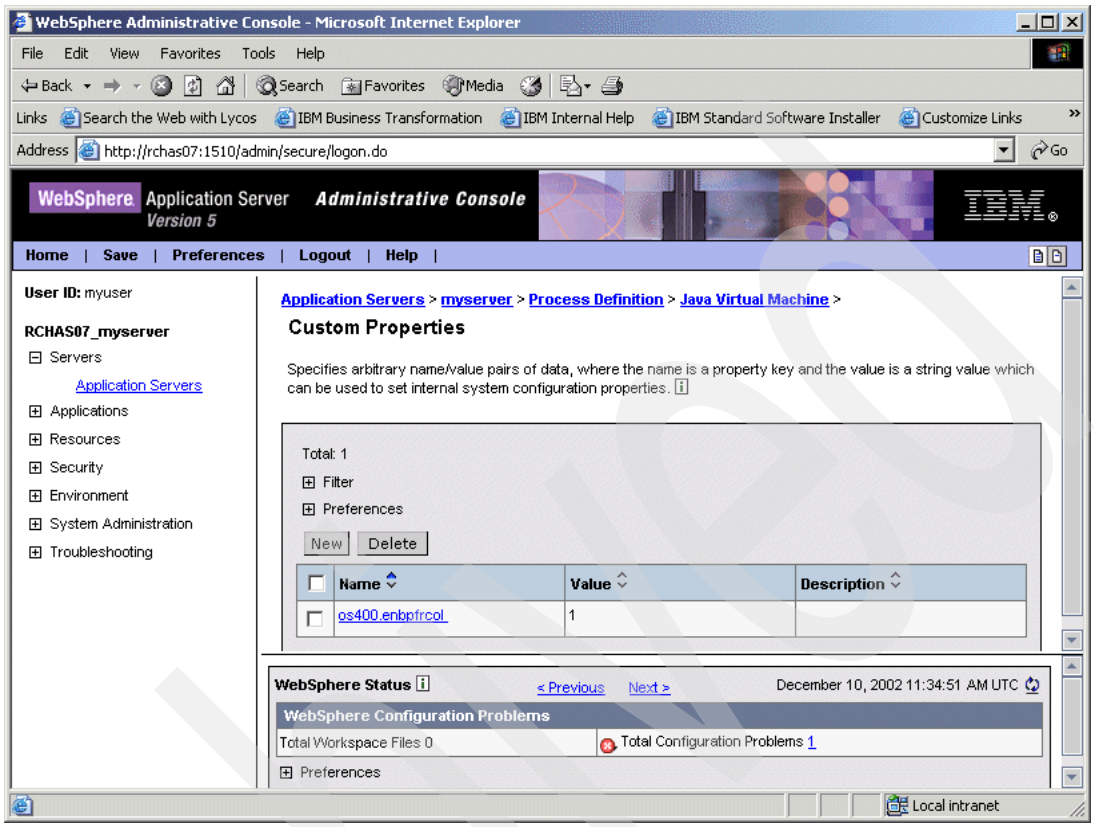


Figure 11-47 Setting WebSphere Application server to collect performance data

b. Start PMI services in WebSphere Application Server. Figure 11-48 shows an example of the administrative console:

i. Expand **Servers** in the left column.

ii. Click **Application Servers** in the left column.

iii. Click the server for which you want to enable performance monitoring.

iv. Click the **Configuration** tab.

v. Click **Performance Monitoring Service**.

vi. Select **Startup**.

vii. Select custom for the Initial specification level. We will make a trace only for:

systemModule=H

threadPoolModule=H

transactionModule=H

viii. Click **Apply** or **OK**.

ix. Click **Save** to save the configuration.

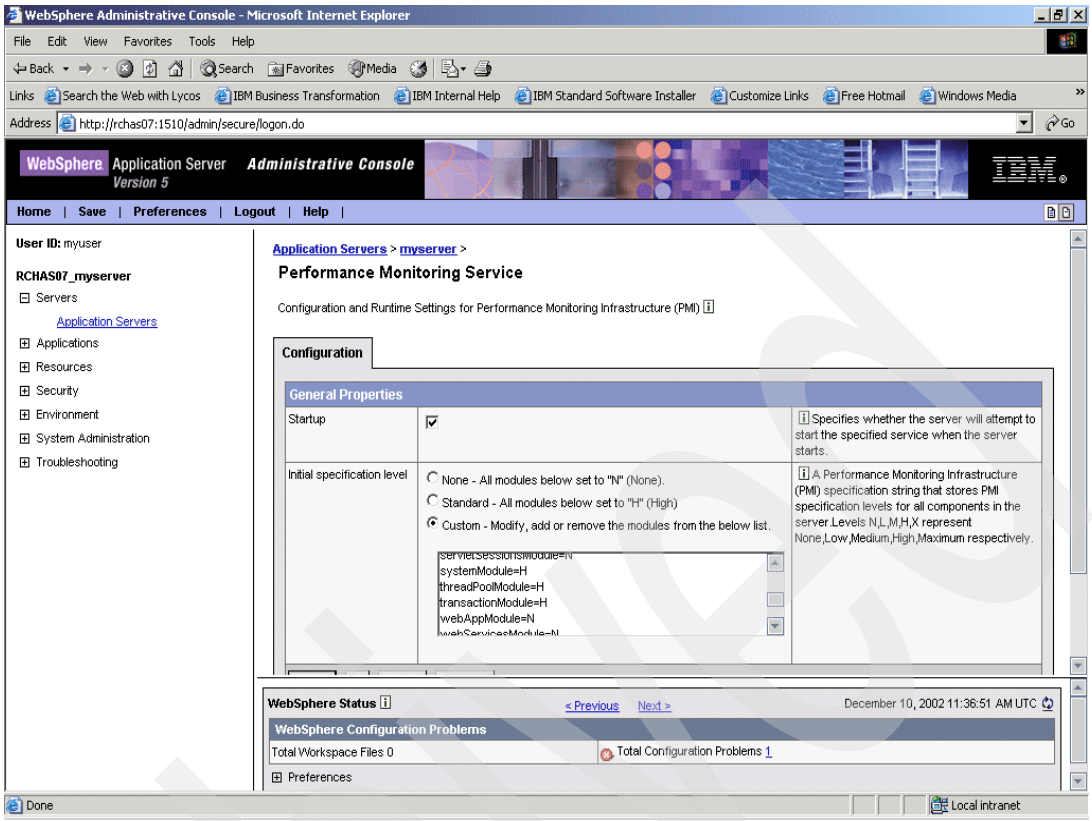


Figure 11-48 Setting Performance Monitoring Service

2. Restart the application server.
   The changes made do not take effect until you restart the application server.
3. Collect performance for analysis:
   a. From an OS/400 command line, type `WRKACTJOB SBS(QEJB5)` and look for your instance's job. Write down its name and user. In our example, the instancename is `MYSERVER` and the user profile is `QEJBSVR`.
   b. Type the following command to create a new PEX definition:

      ```
      ADDPEXDFN DFN(TRACE_WAS) TYPE(*TRACE) JOB((*ALL/QEJBSVR/MYSERVER))
      MAXSTG(100000) SLTEVT(*YES) JVAEVT((*LCKSTR) (*UNLCK) (*THDNFY) (*THDNFYALL)
      (*THDWAIT)) APPEVT((*WAS))
      ```

   c. Start your PEX collection by typing:

      ```
      STRPEX SSNID(TRACE_WAS) DFN(TRACE_WAS)
      ```

4. Execute your WebSphere Application server application.
5. End your PEX collection trace. You must end your PEX collection as soon as you capture the data that you want to analyze. Execute the following command:

   ```
   ENDPEX SSNID(TRACE_WAS)
   ```

6. Using PTDV to analyze your trace:
   a. Start the PTDV tool from your client workstation.
   b. Select **Thick Client**.

c. Enter the following parameters:

   Server Name: `RCHAS07`

   User ID: `ITSCID16`

   Password:

   PEX Library `QPEXDATA`

   PEX Collection `TRACE_WAS`

d. Click **Start Processing**.

7. The cumulative summary window its shown, and you can work with the detailed information in the other two windows. An example is shown in Figure 11-49.

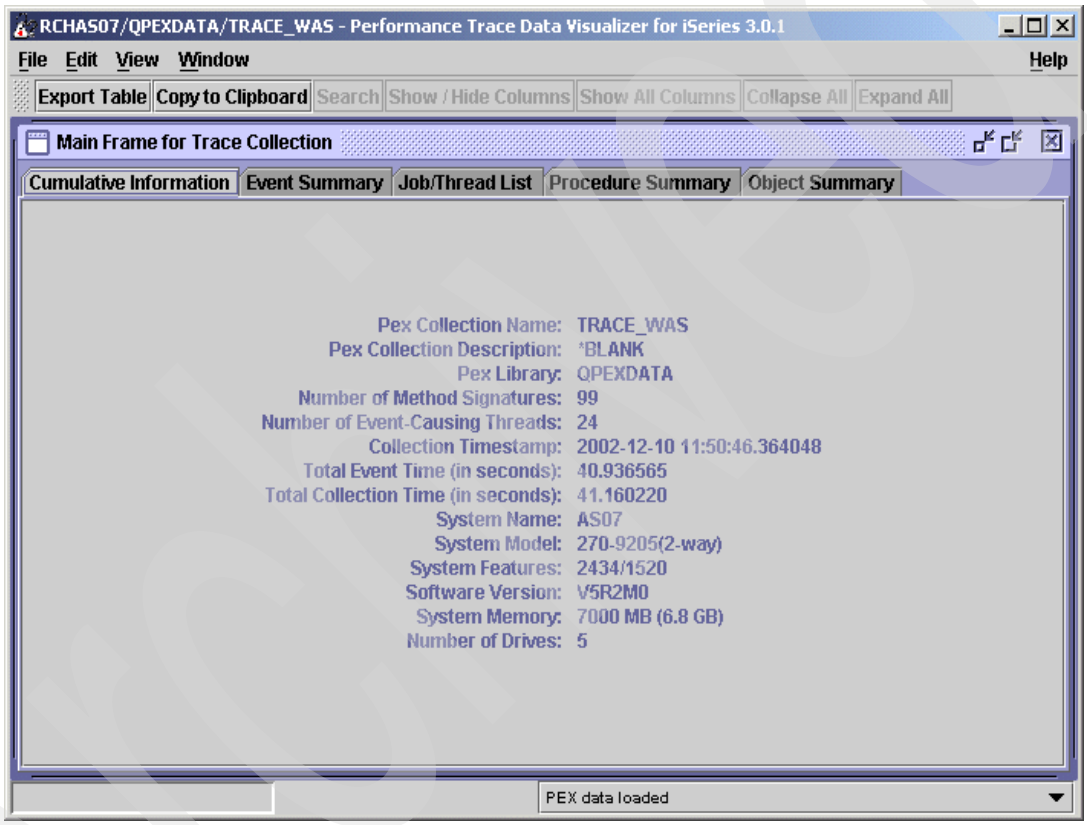


Figure 11-49 Recovering PEX collection by PTDV tool

8. You can navigate through all available windows in PTDV in order to review the PEX collection.

12

Advanced topologies

Choosing an appropriate topology can make a big difference in the long-term success, costs, and performance of your application environment. In this chapter we discuss considerations and implications regarding various alternative network topologies.

What do we mean by topology?

A topology basically refers to a collection of servers and network connections that work together to process the incoming user requests. Some common servers that are required would be an HTTP server, an application server, and a database server. Many other servers may be involved as well, such as DNS servers, LDAP servers, Domino, and message queue servers.

What do we mean by servers?

At the most basic level, a server is an IP address and a port number that follows a defined protocol for receiving and servicing requests. It is important to note that a server is not always a reference to a physical machine. A single machine can serve requests on any TCP/IP port. So, a single machine may be configured to host several servers at the same time. The position of these servers in the network, and the means used to access them, plays a major role in the overall response time of the application.

12.1 Defining the existing topology

Most installations have existing servers in the network which are to be integrated with the WebSphere Application Server. So, it is important to define the existing network topology and which servers we plan to utilize with the applications running under WebSphere Application Server. It is also important to consider if relocation of these servers in the network is an option that may streamline the final configuration.

12.1.1 Identify existing servers

Identify the servers that already exist in the network, and identify which of those are to be used in the planned configuration. The process of integrating these servers is similar to assessing where to position new servers in the network. Some existing servers may not be flexible in how they are accessed, nor where they are located. Because of this, WebSphere is architected to flexibly integrate with other servers in existing networks.

It is important to consider the service times of the existing servers that are to comprise the final solution. If existing servers have long service times, then new servers may require faster than average service times to assure acceptable overall application response times. For example, if an existing slow database server is required by the application, then it may be necessary to use HTTP servers and application servers that are faster than would be required by a topology with a faster database server available.

If there are existing bottlenecks and performance problems in the existing topology, ensure that the workload you are adding to the network avoids them, or add additional capacity to accommodate the new workload. This additional capacity may mean adding server capacity to handle more requests, or network capacity by adding bandwidth or additional routes to reduce contention.

12.1.2 Network alternatives

The choice between ethernet and token ring is also typically a question of the hardware that may already be available in your environment. If the same network medium is used throughout your network, then the latency and expense of bridges can be avoided.

Using optical fibre may be another option. Fibre typically has higher bandwidths available. Another advantage of fibre is an element of physical security. It is not easy to intercept the signal being transmitted nor to physically tap into the line undetected. A wireless network would be the opposite. The signal is easily received, and perhaps even modified, by eavesdroppers, without the knowledge of the sender.

Once a physical medium is selected (or no network, as in the following discussion), you must consider the ease with which the communications may be monitored, and ensure that you plan for protection appropriate for the data. The level of protection is different for a sales catalog, than for plans on how to build a military aircraft. We now discuss the networking alternatives, with their relative performance levels and abilities to protect the transmitted data.

No network

If all the work can be done in the same physical system, then no network is required between back-end servers. For example, if the HTTP server, application server, and database server are all on the same system (as shown in Figure 12-1), then the only network connection is out to the client. But the connections between the servers would all be achieved with no network. A no-network topology is the **best performing alternative**, because there is no physical network and no network latency. Because data is not flowing on a physical wire, encryption is not required to protect it from eavesdroppers, so no-network is the **most secure alternative**.

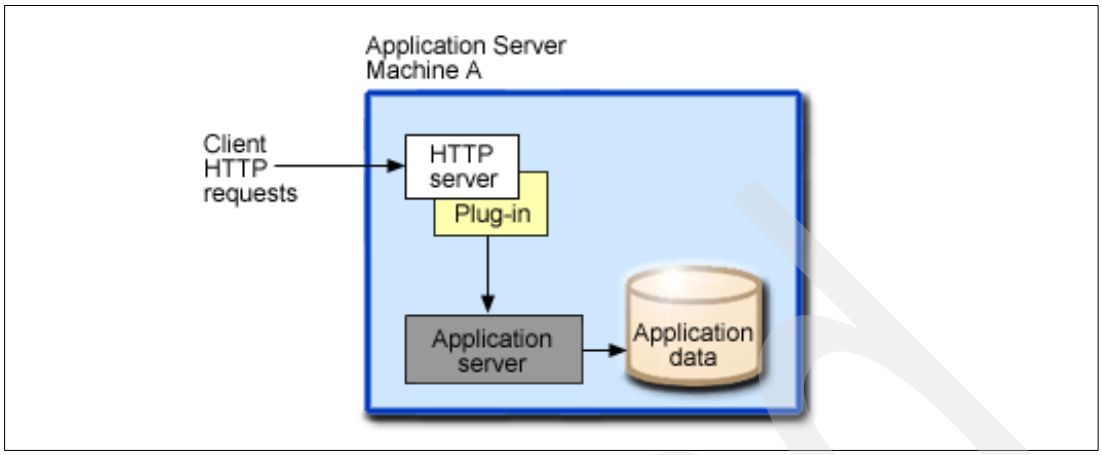


Figure 12-1 No network topology

For example, when choosing where to place the database server in your network, you should try to position it “close” to the applications that most use the data. If you can place the database on the *same* system as the applications, then you can use the *native* JDBC driver. This means using the JDBC driver which calls the database code directly, with no TCP/IP involved at all. The entire database request is processed within the application job. No communication is necessary for the application to access the database.

JDBC is not the only iSeries host server with native optimizations to perform calls directly into OS/400 rather than using sockets. These optimizations are available when running in the iSeries JVM. For more information, see the native optimizations portion of the Toolbox for Java FAQ:

`http://www-1.ibm.com/servers/eserver/iseries/toolbox/faq.htm#nativeOp`

Important: Communications between logical partitions in an **LPAR** environment are *not* the same as the no-network topology described above. Communications between partitions is performed on a virtual LAN at high bandwidth. Communications traces may be performed on this virtual LAN and thus cause security exposures which are not present in the no-network topology. Also, direct calls to services such as to the database via the native JDBC driver are not possible. For these reasons, an LPAR environment is conceptually the same as physically separate systems in the network. By default, communications between LPARs would be like the unprotected network described below. Encryption may still be desirable, performance is still not as good as a no-network topology, and network latency is still present (although low).

Unprotected network

An unprotected network is said to pass data “in the clear”. This means that data is not encrypted. It is clearly visible on a communications trace, and anyone that is able to monitor the traffic on the communications medium used can understand the data. An unprotected network is faster then the more secure methods discussed below, and has low CPU overhead.

In general, if the data flowing back to the browser client is encrypted (HTTPS), then any connections between servers moving any of the response data should also be encrypted. And therefore should *not* use an unprotected network topology. This primarily pertains to the data flow to and from the database server, but also keep in mind that portions of data may also flow to Message Queue Manager, data queues, other sockets servers, RMI methods or remote program calls (see Figure 12-2). Using the alternatives discussed below, you may need to secure all of these network connections to adequately protect the data.

With an unprotected network, data must physically be moved between systems (or partitions) both to send the request to the server, and receive the response. This means it could easily be traced or monitored by eavesdroppers.

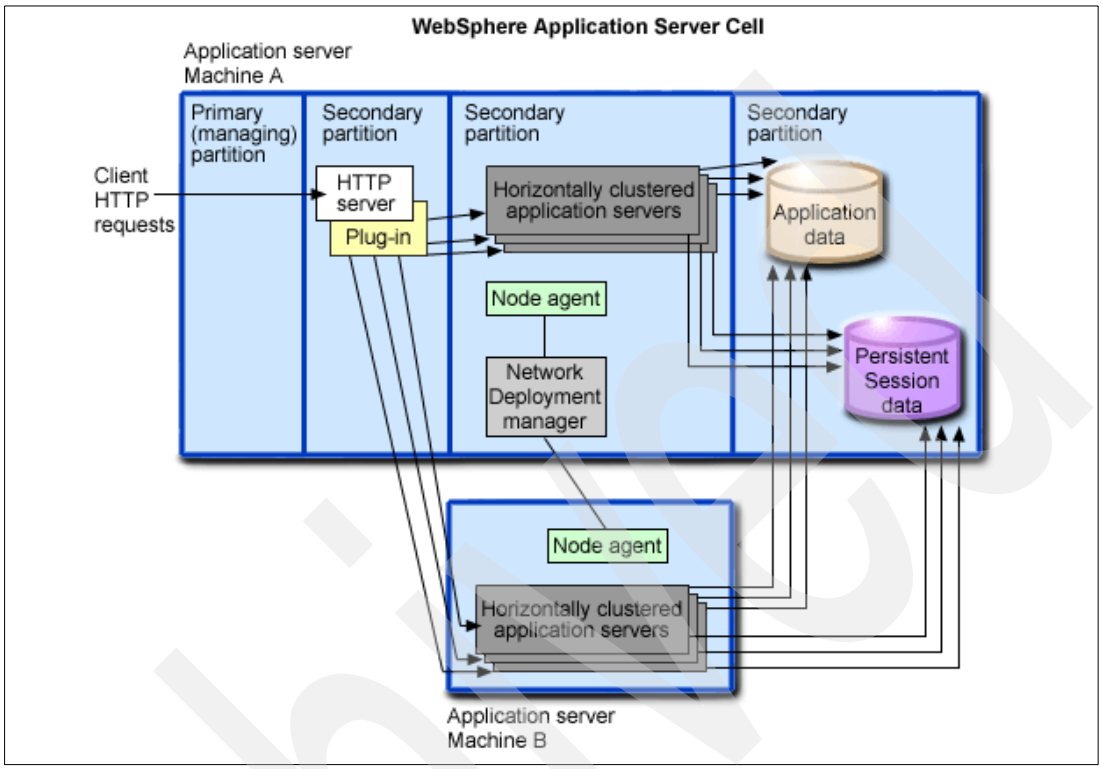


Figure 12-2 Unprotected network topology

Private network

A private network is one that is physically committed to specific servers. Leased lines and local LANs are examples of private networks. A private network is an unprotected network, with physical security around it. That is to say that the data flowing on the network is not encrypted, but steps have been taken to assure that the data won't be visible to anyone without authority. With the broad use of TCP/IP throughout most networks, private networks are less common than they were prior to the availability of the internet, where the network is shared.

Leased lines have always been subject to traces by unauthorized parties. And many networks now extend beyond the enterprise. It is also very difficult to assure you have adequate physical security. The need to avoid these potential security exposures of private networks is why VPN was developed (see below).

A private network is one that is physically secured and controlled. A private network is presumed to be secure from eavesdroppers, and therefore may be used to pass data "in the clear". Because it is physically secure, and does not require encryption, a private network can give you greater performance and less overhead than SSL or VPN. Performance is the same as the unprotected network topology described above, because from the system's perspective, the two are the same.

In an LPAR environment, the virtual LAN used to communicate between partitions is an example of a private network. In such an environment it is important to control access to the communications trace functions of the system. Any user with the ability to trace could see data that they are not intended to access.

SSL / TLS

The Secure Sockets Layer (SSL) and Transport Layer Security (TLS) — see RFC 2246 at http://isc.faqs.org/rfcs/rfc2246.html) — use encryption of the data that is transmitted. While anyone can see the transmitted bytes, they are not the actual data. To determine the actual data, decryption with the proper conversation key must be performed. SSL differs from VPN (discussed below) in that SSL may be enabled, or not, for *each connection*. Whereas VPN is configured on the network interface and encrypts all connections using that interface.

Note: Encryption does *not* mean the data cannot be seen by an unauthorized person. In fact, there are published estimates of how much time and money it would take to "attack" keys of various strengths. So, if the data is of sufficient importance to motivate such an attack, it should also have physical controls placed on it. For example, don't transmit diagrams on how to assemble a military aircraft over a wireless LAN.

There is significant CPU overhead involved to establish SSL *connections* between the servers. In environments such as HTTPS secure Web sites, there are large numbers of secure connections being established, and they typically only last a short time. In such an environment the addition of a cryptographic co-processor can dramatically reduce the burden on the system's CPU because 90% of the overhead associated with establishing the connection is off-loaded to the co-processor.

It is also important to keep in mind that there is significant CPU overhead involved in encrypting the *data* that flows between the servers. **Encrypting the data flow requires significant CPU time**. In fact, when compared to an unprotected TCP/IP network conversation, an established SSL connection, may require more than 8 times as much CPU to encrypt the data send on the network adapter.

In environments where there are only a small number of SSL connections made, the cryptographic processor does not offer a great benefit to overall performance. This is because the co-processor aids with establishing the connections, but the encryption of the data is performed by software on the system's CPU. An example of such an environment is a JDBC connection to a database server (such as is shown in Figure 12-5 on page 514). Typically a small number of connections are used for long periods of time. Therefore, a cryptographic co-processor would not improve overall performance of such an environment.

For more information on encryption hardware, see the following section of the V5R2 InfoCenter:

Networking > Networking security > Cryptographic hardware > Cryptographic concepts

Note: Using SSL over an encrypted VPN (discussed next) would be redundant. Such a configuration would incur the overhead of encrypting all of the data at two levels rather than one, and would not offer any significantly greater security.

VPN

Virtual Private Networking (VPN) is a reference to a dataline, or network connection that resembles the private network topology. A public network may be used to support the conversation, but all connections on a VPN are authenticated and may also be encrypted. In a theoretical sense, because others cannot send valid, authenticated data on this connection, VPN resembles a leased line, where the two endpoints have their own wire between them. When encryption is configured, and others cannot monitor the data being exchanged, then the network is actually more private (that is, less traceable) than an actual private network.

VPN may be configured to perform authentication only, or authentication *and* encryption. Authentication is accomplished using the Authentication Header protocol, and encryption is performed with Encapsulating Security Payload.

Authentication Header protocol

The Authentication Header (AH) protocol provides:

- Data origin authentication
- Data integrity
- Replay protection

AH does *not* provide data confidentiality, all application data is sent "in the clear". To encrypt transmitted data, configure VPN with Encapsulating Security Payload (ESP). If you are just trying to ensure that the data came from your other system, and that an unauthorized party is not trying to send requests to your server, then using an AH-only configuration may be appropriate. This is the *data origin authentication* capability of the AH protocol.

AH provides *data integrity*, because information about the data is stored along with the authentication information. If any of the bits in the data or the authentication information are altered, or mistransmitted, this inconsistency is detected (with high degree of probability anyway), and the datagram is discarded.

Replay protection is a reference to someone attempting to re-send datagrams. If the frame was authenticated and accepted the first time it was transmitted, an attacker might attempt to send a copy of the datagram a second time, or even thousands of times. If a second copy of your bank deposit record is processed, that would be a problem (for the bank anyway). The replay protection imposes a sequence integrity over the communicated datagrams as well. Thus, an identical datagram would not authenticate a second time, because it would not follow the proper sequence.

For the best performance, configure the lowest level of security demanded by your application. While ESP also performs authentication, **AH-only affects system performance significantly less than ESP**. AH-only also authenticates the *entire* datagram, whereas ESP, does not authenticate the leading IP header or any other information before the ESP header. Packets that fail authentication are discarded and are never delivered to the recipient. This greatly reduces the chances of successful denial of service attacks.

Encapsulating Security Payload (ESP)

OS/400 VPN-ESP may use the following encryption methods:

- Data Encryption Standard (DES)
- Triple-DES (3DES)
- RC5
- RC4
- Advanced Encryption Standard (AES)

Tip: If possible, take advantage of the performance improvements offered by RC4 rather than DES VPN encryption. Using ESP with RC4/MD5 encryption can have twice the capacity and half of the CPU time of ESP with DES/MD5.

For more information on VPN see:

- The V5R2 InfoCenter:

  Networking > Networking security > Virtual private networking

- Chapter 5, "Network performance", in *iSeries Performance Capabilities Reference V5R2, June 2002 Edition* SC41-0607 at:

  http://publib.boulder.ibm.com/iseries/v5r2/ic2924/books/as4ppcp5.pdf

12.2 Criteria for evaluating topology alternatives

With so many networking alternatives, each with their own characteristics, how can we decide on a combination that is appropriate for our environment. In this section, we present some of the key criteria that are used to weigh the alternatives.

12.2.1 Security

You must plan the topology around the required level of security. Review the existing controls over the access to the application and its data to help define the security requirements of the new topology.

In general, the entire network should be secured to the extent necessary to protect data from employees that are not otherwise authorized to see it. For example, if the network staff is not authorized to the application data, then steps should be taken to secure the data traveling on the network. On the other hand, it may not make sense to take on the burden of encrypting information on your internal network. If the building is considered physically secure, and all employees have the same access to application data, then encryption on internal networks doesn't offer much additional security for its performance impact.

The ideal topology:

- Delivers the required level of security with very low risk of intrusion
- Has acceptable response time
- Has reasonable cost

12.2.2 Data integrity

Some environments require redundant copies of data in the network. This can guard against data loss in the event of a disaster at any one physical location. However, managing the data and insuring the correct data is returned from the database is obviously essential. When multiple copies of data exist, steps must be taken to assure data integrity.

Also, it is important to keep in mind that data transmission is not 100% reliable, and data may become corrupted on the network. It is also possible for someone to attempt to alter datagrams as they flow through a network. A VPN network may be appropriate to address several of these concerns at the same time.

If only one copy of the data exists, then protecting the system that hosts it is essential to the availability of the application. For example by using battery backup, journalling, commitment control, RAID-5 or mirroring protection of secondary storage, and a thorough backup and recovery plan.

If the network that transports the data is not physically secure, or you want further assurance that the data has not been tampered with, a VPN (see discussion above) further assures data integrity.

The ideal topology:

- Assures delivery of accurate data from a highly available database server
- Is simple to maintain and configure
- Has a cost lower than costs of an outage

12.2.3 Topology performance

Keeping the data private and secure often means using encryption. Every time data is sent across the network it must be encrypted again. This can quickly lead to degraded service times. If related servers can be positioned on the same host, then it may not be necessary to encrypt data between these servers. For example, the native JDBC driver, and the native implementations of the Toolbox for Java perform calls directly to the function they wish to perform, rather than using a network to interface to a host server.

A server always has a software component to it. Often this software is part of an extensive array of middleware that comprises the environment. Using software with fast service times for back-end servers improves the performance of the overall topology.

The hardware running the server software impacts performance, as well as availability of the application. For servers that are required to process most all of the transactions (for example a database server), it is important to assure high availability. In general, such **single points of failure also prove to be points of contention as transaction rates increase.** Therefore, you can often improve both availability and performance with redundant servers.

Network performance and throughput can become an issue when large volumes of data are being moved across the network between servers, or out to the end-user. Prototyping the bandwidth requirements between servers is helpful when the servers are on different systems throughout the network.

The ideal topology:

- Delivers acceptable performance at times of peak demand
- Is scalable to address performance bottlenecks as usage increases

12.2.4 Application availability

One of the primary goals of complex topologies is to provide higher levels of system availability, and disaster planning. When considering a network topology, you should consider the following factors of application availability:

- Network hardware
- Server hardware
- Middleware
- Application software
- Natural disaster

Your application's availability is only as good as the weakest link in the topology. If three servers are all required to service an application request, and each of the three is on a separate piece of hardware, a failure on *any* of the servers makes the entire application unavailable. Adding redundant servers for those weakest links, with higher expected failure rates, may preserve the availability of the application.

Consolidating servers onto fewer machines and then taking steps to protect those machines may provide better availability than having large numbers of machines, each running a specific server.

While having all the servers on as few systems as possible simplifies configuration and management, it also creates a disaster exposure. Even if the servers are on separate systems, if the systems are all at the same physical location, you must consider the course of action that would be required in the event of an extended power failure, flood, tornado, earthquake or other disaster.

Your disaster plan could be as simple as restoring a recent system backup to another system, or as complex as a real-time data backup, with automated failover support. You should consider the time, expense and likelihood of implementing your disaster plan, whether a single system to host multiple servers simplifies the disaster scenario, and whether a single system makes it any more or less likely that you have to implement the plan.

With iSeries servers, availability of a single system may be enhanced by utilizing combinations of the following features:

- Disk protection: RAID-5 or mirroring
- Hardware And Operating System Redundancy: Lpar
- Failure of network adapter or carrier: multiple network interfaces

The ideal topology:

- Is available 100% of the time
- Or at least 100% of the time the users expect it to be available
- Or is highly available, with most outages occurring on a scheduled basis
- Is resistant to disasters at any one location with minimal time to activate an alternative server configuration

12.2.5 Adaptability to evolve

When choosing a topology, you must keep in mind that your business is not the same now as it was ten years ago. And it will not be the same ten years from now. So it is important to have a topology that is adaptable to changes in the architecture and scale of your business. Technologies and standards are also evolving. The relative cost of bandwidth is dramatically different now than just a few years ago.

The type of data being served may change over time. This may require more stringent security than the initial configuration, or higher bandwidth. By architecting an adaptable topology, you can more easily accommodate such potential changes when they become necessary.

The ideal topology:

- Adapts easily to changing application requirements
- Adapts easily and inexpensively to resolve performance bottlenecks

12.2.6 Hardware platform

If the software you have chosen for specific server functions is only available on specific hardware platforms, then this obviously limits your options. However, the iSeries has many features which may relieve some of these limitations. The iSeries supports Linux partitions, Integrated Netfinity® Servers (which run Windows-based applications), and the Portable Application Solutions Environment (PASE) (to run many pSeries (AIX) applications). So, even in cases where the server software does not specifically state support for iSeries, it may still be possible for the iSeries to host the server.

12.2.7 Simplicity

If the architected topology is too complex to manage, this is typically problematic for the runtime environment. Complexity inherently brings with it unexpected failures and less flexibility to adapt to change. Ironically, attempts to insulate the application from failure often introduce complexity.

For example, to insulate ourselves from the risk of a hardware failure on the database system, we incorporate a backup system. If the backup system is ever going to be useful, then we must also architect the failover process and have both database systems available to the application, and a means for the application to assess which is the proper database host. This is quite complicated when compared to the single database host scenario. And by having two systems running rather than just one, we've also increased the likelihood that one of them encounters a hardware failure.

Extending the concept of redundancy to servers for HTTP and load balancing, firewalls, LDAP, SOAP, Domino, etc., makes it clear how important it is to keep things simple — and also demonstrates how simplicity often has a lower cost.

Estimate the ongoing costs of complexity and weigh them against the costs of allowing the application to be unavailable. There are many cases where the complexity is not worth the minimal improvement in application availability.

The ideal topology:

- Adapts easily to changing application requirements
- Is managed centrally by a single administrator

12.2.8 Ongoing support costs

Complex environments can avoid downtime, but they can also *cause* downtime (as discussed above). The cost of installing, servicing, and later upgrading the components in the topology are all factors that must be considered. Initial and ongoing costs of software licenses and hardware maintenance should also be considered.

The ideal topology:

- Has reasonably priced software and hardware maintenance
- Does not require extensive administration

12.3 Sample topologies

Often, when planning your network, you have choices on where to place a given function. database servers, LDAP servers, HTTP servers, application servers, Java beans, Web services and other TCP/IP-based servers can all, in theory, be placed anywhere in the world that you can reach with a TCP/IP connection.

In general, you can place a given server on an existing system, a new system or a new logical partition (LPAR). In the following examples topologies, we compare and contrast several common topologies.

12.3.1 Single system topology

Hosting all of the servers on a single system has many advantages. There is only one system to keep running, and maintain. There are no network delays between servers as they interoperate and no encryption required to protect data exchanged between the servers (see Figure 12-3).

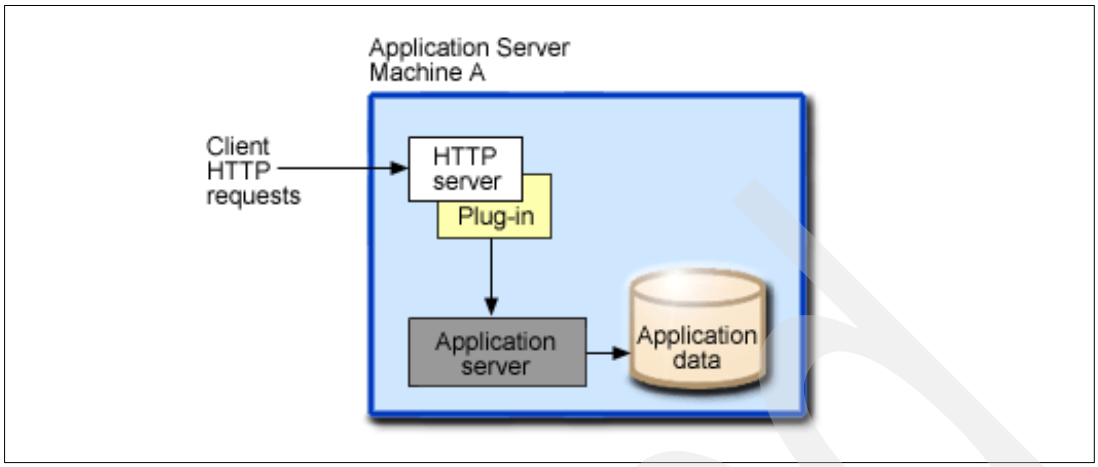


Figure 12-3 Sample topology - single system

This diagram shows an example of a single system environment. Notice that the client requests are not done via HTTPS and so are done “in the clear”, and are easily observed by anyone tracing the network connection.

In this topology, we have rated these aspects as follows:

- **Security:** Poor, data flows in the clear, and the server is not protected from attack by a firewall. We suggest not starting any TCP/IP servers other than HTTP in this environment. We also suggest enabling denial of service attack features in the HTTP server, and configuring the application server to *only* accept connections from the IP address of machine A.
- **Data integrity:** Good, there is only one copy of the data, and no network to allow tampering with data transmissions. We suggest taking action to insulate the environment from a disk failure by using RAID-5 or mirroring. We also suggest a thorough backup and recovery plan with off-site storage of data.
- **Performance:** Average, there are no delays due to encryption, or interfacing with other servers. With a single system baring the entire workload, the ability to scale the application to higher activity levels may be limited, but all the functions of the application benefit from any hardware expansion. We suggest choosing an iSeries model which supports your expected workload before approaching a maximal configuration. This leaves room for expansion within the original system model, as workloads increase.

Attention: You may have seen other discussions that indicate a single application server, HTTP server or database server does not perform as well as a cluster of servers or separate servers. While it is true that one system may become a bottleneck when it is pushed beyond its capacity, many of the other performance reasons for clustered environments are less pertinent to iSeries systems. Although the failover support is still a strong point for those environments.

The iSeries JVM runs continuously without halting for garbage collection. Threads of a job may run simultaneously on separate CPUs. Main storage is shared across all the active threads. In general the iSeries operating system (OS/400) and the JVM are built to efficiently multi-task, and utilize system resources at high levels of concurrency.

Before planning a more complex topology, go back to the evaluation criteria and make sure the complexity, security exposure and operating expense are worth that last small amount of availability.

- **Availability:** Average, since there is only one system to risk failure, we don't multiply the risk by multiple server systems. However, with no redundancy in the configuration, any failure would take the application off-line. We suggest implementing RAID-5 or mirrored disk protection, along with battery backup. You should also consider how to accommodate installation of software updates, or hardware expansion. In this environment, these events require taking the application off-line. If the application is not expected to have 24 hour availability, then this limitation may be acceptable. The sacrifice of availability during scheduled upgrades may be well worth the savings resulting from less server systems, and less complexity.
- **Adaptability:** Good, because we can always introduce new components as needed. For example, there is nothing in this configuration to complicate the task of moving the HTTP server to another system, or to add a logical partition to maintain a redundant copy of the database.
- **Simplicity:** Excellent, with only a single network connection involved (between the server and the client), it is straightforward to determine if a failure is due to the network, a server application, or a hardware failure. To help an administrator make such an assessment, we suggest planning for some means to administer the environment other than the network connection. Perhaps using a dial-up PPP connection as an alternative means of accessing and monitoring the server.
- **Ongoing costs:** Excellent, with only a single operating system license and hardware maintenance agreement, and an extremely simple configuration, ongoing costs should be minimal when compared to other topology alternatives.

This topology may be appropriate in environments where disaster planning and data protection are not critical. For example, if a company only has one location, and a flood disables this single system application, the server being down is not likely to impair their business any further than the flood has already done.

12.3.2 Demilitarized zone (DMZ) topology

It is not always possible or desirable to host all of the servers on a single system. It creates a security exposure from network attackers. You may also have requirements to utilize existing server systems on your network. A DMZ topology adds two firewalls, and an HTTP server to the single system topology (see Figure 12-4). Typically the first firewall only allows HTTP requests to come into machine A. The second firewall only allows application server requests to come into machine B. This assures that network attackers cannot directly attack the database server.

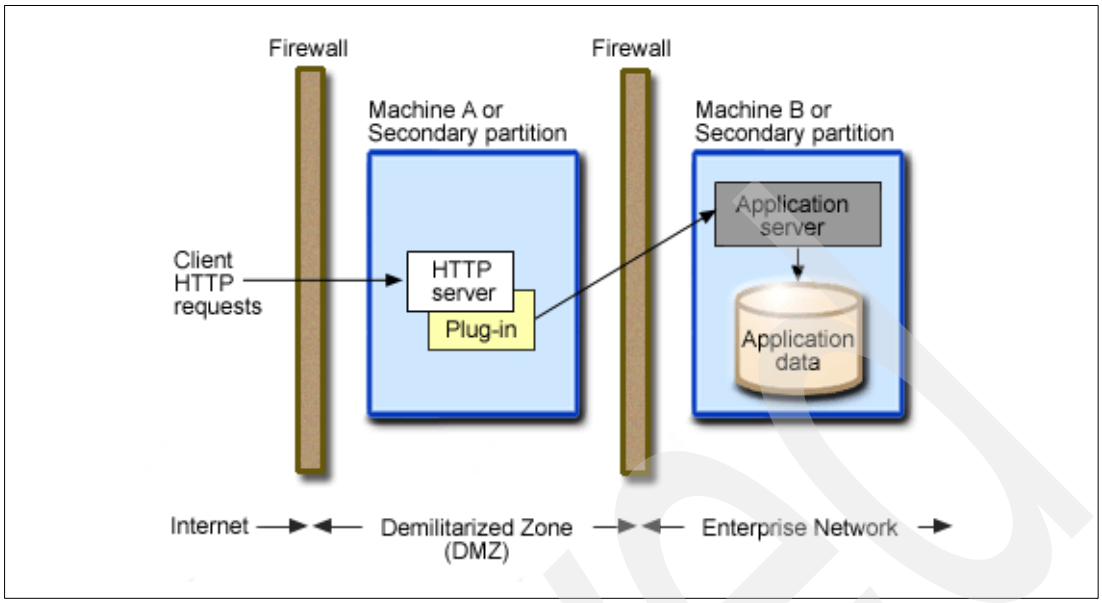


Figure 12-4 Sample topology - demilitarized zone (DMZ)

In this topology, we have rated these aspects as follows:

- **Security:** Good, the firewalls in the topology allow you to safely open TCP/IP access to machine B for other host servers, such as data queues, remote command calls, and other functions for use within the enterprise network. The data is still flowing “in the clear”, and therefore is traceable. If the connection between machine A and B is on a private network, it may be adequate to secure the data by using HTTPS connections back to the client.
- **Data integrity:** Good, there is still only one copy of the data and only a private network where any tampering with data transmissions might occur. We suggest the same precautions described in the single system topology.
- **Performance:** Good, since the application server and the application data are on the same system, the application can access data with minimal overhead. It can construct the response, and send it back to machine A and the requesting client. Since the request and response only traverse the network once, the performance impact of the firewalls and separate HTTP server system are not a significant factor on response times. Also, the burden of processing the HTTP requests is off-loaded to machine A, and therefore the application server is the only increase in workload to the existing enterprise system, machine B.
- **Availability:** Average, this topology has taken no steps to achieve any higher availability than the single system topology. In fact, availability is reduced from the single system topology because there are now four hardware components involved in every transaction (machine A, machine B, and the two firewall systems). If any one of the four fails, the entire application is unavailable. The same issues with how to install software upgrades and hardware expansion exist. In fact, you now have four systems that require periodic upgrades, rather than the single system topology’s one system.
- **Adaptability:** Good, the machine used for the HTTP server or the firewalls could very easily be changed, and new components to the topology may still be introduced. In this topology, if machine B is undersized for the existing enterprise network requirements plus the new application serving requirements, there are few alternatives but to upgrade machine B, or to move the application server to a separate server system or logical partition. This is due to the existing enterprise software being dependant upon machine B.

- **Simplicity:** Good, since firewalls don't tend to require much ongoing maintenance, and there are only two network connections to maintain, problem determination is still pretty straightforward. As with the single system topology, we suggest an alternate means for system administrators to access the system so they can diagnose potential network or firewall problems.
- **Ongoing costs:** Good, we've added the need for additional licenses for operating systems and firewalls, but we've off-loaded the burden of HTTP serving to another system, and we've added the security of the firewalls.

This topology may be appropriate in environments where an existing enterprise network is already in place. With proper configuration of the firewalls, the DMZ configuration makes it possible to add an application server to the existing network, while insulating the enterprise data from much of the security risk associated with Web serving.

12.3.3 High availability topology with multiple WAS cells

The multiple WebSphere Application Server cell topology involves setting up multiple WebSphere Application Server cells where a different physical machine hosts each cell. Applications are deployed onto multiple WebSphere Application Server administrative cells (see Figure 12-5). This topology requires WebSphere Application Server Network Deployment.

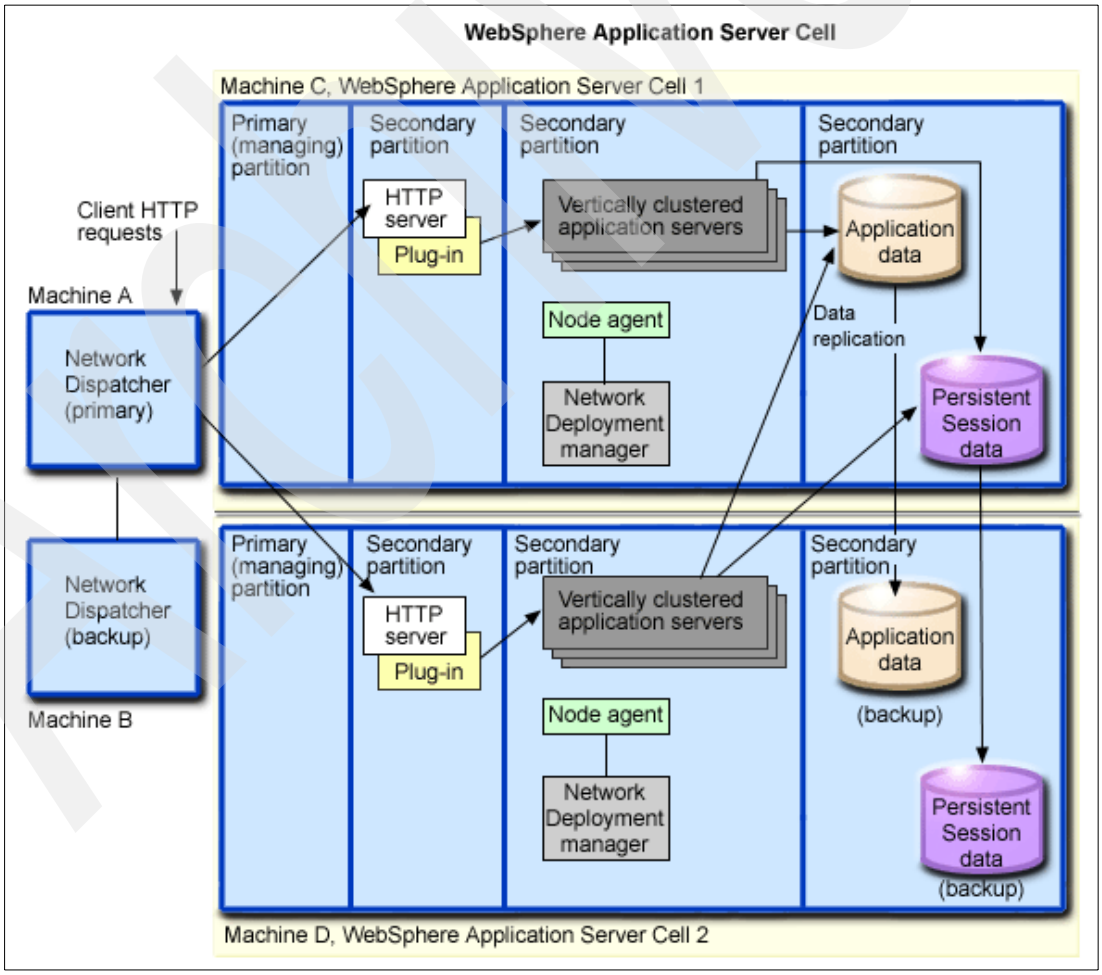


Figure 12-5 Multi-cell topology for high availability

The advantage of this topology is in its support of redundancy at each component: Network Dispatcher, WAS, HTTP server, and database. This topology includes the following features:

- Machines C and D host two separate WebSphere Application Server cells. Identical versions of an application are deployed in both cells.
- To ensure that identical versions of the application run in both cells, the application server cluster members in both cells are identical to each other.
- You administer each cell separately. Each cell has its own set of XML configuration files and its own Network Deployment manager.
- Both machines use logical partitioning (LPAR) for data and process isolation.
- Machine A hosts the primary Network Dispatcher node. This machine distributes incoming HTTP requests to the two cells. Network Dispatcher presents a single image of the application to the clients. A backup Network Dispatcher node (Machine B) provides failover support.
- This topology also shows the application database and persistent session database. Both cells in the topology share a common application database. The databases use data replication to enhance availability and avoid creating a single point of failure.

It also is possible to run a different version of the application in each cell. For example, you can create a test environment in one cell and a production environment in the other cell. Because the cells are isolated from one another, you can also run different versions of the WebSphere Application Server software in each cell. However, this layout undermines the goal of this topology: highly available solution running WebSphere applications.

Note: Running different versions of the application or WebSphere Application Server does not ensure high availability. To ensure availability in a multiple cell topology, you must configure at least two of the cells to run identical versions of your application.

In this topology, we have rated these aspects as follows:

- **Security:** Good, because data transmission is primarily over a private network. We recommend the two cells not be in the same physical location for disaster planning, therefore a VPN or SSL connection between cell 1 and 2 would be highly recommended.
- **Data integrity:** Excellent, due to the inability to alter data frames flowing on the virtual private network between the partitions, and the same protection when replicating to the backup database by using VPN.
- **Performance:** Good, this is achieved by:
  - Running an application in multiple smaller cells. It can provide better performance than a single large cell because there is less interprocess communication in a smaller cell.
  - Using one or more of these workload management techniques:
    - Use the WebSphere Application Server Network Deployment workload management (WLM) facility to distribute work among application server cluster members
    - Use Network Dispatcher to distribute client HTTP requests to each Web server

    For example, an application can manage workloads at the Web server level with Network Dispatcher and at the application server level with WebSphere workload management. Using multiple workload management techniques in an application provides finer control of load balancing.

- **Availability:** Excellent, due to the many redundant servers available in the network. When you deploy an application in two or more cells, any problems that occur within one cell are isolated, and the other cells continue to handle client requests. Process isolation allows you to maintain high availability in any of these situations:
  - When you deploy a new application or a revision of an existing application. You can test the new application or revision in one cell while the other cells continue to process client requests.
  - When you deploy a new version of WebSphere Application Server for iSeries. You can test the new version in a live environment without interrupting service.
  - When you apply fixes or patches to WebSphere Application Server for iSeries. You can perform hardware and software upgrades on a cell-by-cell basis during off-peak hours to avoid an interruption of service.

    If a problem occurs with the new software, you can accomplish a rollback to a previous software version more quickly.
- **Adaptability:** Limited, due to the significant infrastructure already in place. However, the backup cell does afford the ability to bring a new version of the application on-line without disrupting the existing application.
- **Simplicity:** Poor, due to the requirement of maintaining a large number of hardware and software components.
- **Ongoing costs:** High, due to the large number of server systems, and many copies of various middleware components.

This topology may be appropriate in environments where high availability to the application is essential, and where peak application demand may exceed the capacity of a single cell.

12.4 Summary

You may have noticed that most of the sample topologies were considered quite adaptable. This is part of why TCP/IP and other standard protocols are becoming so important in the marketplace. But you should notice that in order to improve on one of the topology criteria, there is generally a trade-off in another.

If you are considering a change to improve one element of a given topology, and have not found *any* trade-off in another, then you probably have not fully considered the implications of the change. Higher availability will almost always increase costs and complexity. Simple topologies will be sacrificing security or availability. Secure topologies will have impaired performance. And so forth.

The key is to match your planned topology with your *most important* criteria. If security is paramount, then you may have to allow for longer response times to be within the project's budget. If cost is paramount, then it is important to assure the potential security exposures are sufficiently understood before implementing the topology. If high availability within a fixed budget is the goal, then perhaps allowing for longer response times during unplanned outages would be a reasonable compromise.

A

Reference Desk

This appendix provides a quick reference to the information you will find handy to have available as you are administrating and troubleshooting your system environment. It is organized by the different types of tasks you will be performing.

A.1 OS/400 object names

This section will help you in determining what iSeries objects are created or used by WebSphere Application Server.

A.1.1 IFS path names

This section provides information about most useful path name for WebSphere Application Server V5.0 for iSeries installation.

A.1.1.1 Base version

Product directories:

- Product root directory:

  `/QIBM/ProdData/WebAS5/Base`

- QShell scripts:

  `/QIBM/ProdData/WebAS5/Base/bin`

- Embedded Publish and Subscribe messaging root product directory:

  `/QIBM/ProdData/WebAS5/wemps`

User data directories:

- User data root directory:

  `/QIBM/UserData/WebAS5/Base`

- Embedded Publish and Subscribe messaging user configuration and files root directory:

  `/QIBM/UserData/WebAS5/wemps`

- WAS instance:
  - Root directory:

    `/QIBM/UserData/WebAS5/Base/<instanceName>`

  - Configuration files directory:

    `/QIBM/UserData/WebAS5/Base/<instanceName>/config`

  - Installed applications:

    `/QIBM/UserData/WebAS5/Base/<instanceName>/installedApps`

  - Log directory:

    `/QIBM/UserData/WebAS5/Base/<instanceName>/logs/<serverName>`

  - Property files directory:

    `/QIBM/UserData/WebAS5/Base/<instanceName>/properties`

  - WebSphere plugin configuration file:

    `/QIBM/UserData/WebAS5/Base/<instanceName>/config/cells/plugin-cfg.xml`

A.1.1.2 Network Deployment version

The Network Deployment edition is installed in a different directory.

Product directories:

- Product root directory:

  `/QIBM/ProdData/WebAS5/ND`

- QShell scripts:

  `/QIBM/ProdData/WebAS5/ND/bin`

User data directories:

- User data root directory:

  `/QIBM/UserData/WebAS5/ND`

- WAS ND instance:
  - Root directory:

    `/QIBM/UserData/WebAS5/ND/<instanceName>`

    By default, the name of the Deployment Manager server is *dmgr*.
  - Configuration files directory:

    `/QIBM/UserData/WebAS5/ND/<instanceName>/config`
  - Installed applications:

    `/QIBM/UserData/WebAS5/ND/<instanceName>/installedApps`
  - Log directory:

    `/QIBM/UserData/WebAS5/ND/<instanceName>/logs/<serverName>`
  - Property files directory:

    `/QIBM/UserData/WebAS5/ND/<instanceName>/properties`

A.2 Subsystems

The following is the list of the iSeries subsystems used by WebSphere Application Server V5.0 for iSeries Base and Network Deployment editions:

QEJBAS5	WebSphere Application Server V5.0 for iSeries, Base Edition
QEJBASND5	WebSphere Application Server Network Deployment V5.0 for iSeries
QMQM	Message queue manager
QUSRWRK	The subsystem where the JDBC toolbox driver's jobs are running
QSYSWRK	The subsystem where the JDBC native driver's jobs are running

Visit the following Web site for a more complete list of host server jobs and subsystems:

`http://publib.boulder.ibm.com/iseries/v5r2/ic2924/info/rzaks/rzaksserverjobcheattable.htm`

A.3 User profiles

QEJB: Profile used only for backward compatibility. This user profile is used only when accessing validation list objects used for storing the encoded passwords used with WebSphere Application Server.

QEJBSRV: Profile used by all server process by default except for MQ processes. It also owns directories and files used by WebSphere Application Server. To run an application server under a specific user profile (not QEJBSVR) see instructions at:

`http://publib.boulder.ibm.com/iseries/v1r1m0/websphere/ic2924/index.htm?info/rzaiz/50/sec/isprfchg.htm`

A.4 JDBC information

Most applications use a database to store the application data. You can run a database on many platforms and still access it from an application running in WebSphere Application Server. This section covers some implementation details for accessing a database on iSeries.

A.4.1 JDBC drivers

There are 2 JDBC drivers to access database on iSeries:

- Native JDBC driver:
  - Class name to register: **com.ibm.db2.jdbc.app.DB2Driver**
  - Path to the driver classes: **/QIBM/UserData/Java400/ext/db2_classes.jar**
  - The URL subprotocol: **db2**
  - Host server job:
    - Subsystem: **QSYSWRK**
    - Jobname: **QSQSRVR**
    - Job description is same as QUSER user profile: **QDFTJOBD** as shipped
  - JDBC driver type: **Type 2**; **JDBC 2.0**.
  - TCP/IP port: none. The native JDBC driver calls the database via the SQL CLI.
  - FAQs Web sites:

    http://www-919.ibm.com/servers/eserver/iseries/developer/jdbc/Faqs/JDBCFAQ.html
    http://www-1.ibm.com/servers/eserver/iseries/db2/clifaq.htm
- Toolbox JDBC driver:
  - Class name to register: **com.ibm.as400.access.AS400JDBCDriver**
  - Pathname to the driver classes: **/QIBM/ProdData/HTTP/Public/jt400/lib/jt400.jar**
  - The URL subprotocol: **as400**
  - Host server job:
    - Subsystem: **QUSRWRK**
    - Jobnames: **QZDASOINIT** & **QZDASSINIT**
    - Job description is same as QUSER user profile - **QDFTJOBD** as shipped
  - JDBC driver type: **Type 4**; **JDBC 3.0**, with Toolbox modification level 5 (shipped with V5R2 or JTOpen version 3.x).
  - TCP/IP port: **8471** for QZDASOINIT, **9471** for QZDASSINIT (SSL enabled)
  - FAQs Web site:

    http://www-1.ibm.com/servers/eserver/iseries/toolbox/faqjdbc.htm

Table 12-1 summarizes when to use each JDBC driver.

Table 12-1 iSeries JDBC drivers

Driver name in WAS		Local database access	Remote database access	JTA enabled	OS/400 Version
DB2 UDB for iSeries (Native - V5R2 and later)	IBM Developer Kit for Java JDBC Driver (native Driver)	X			V5R2 and later
DB2 UDB for iSeries (Native XA - V5R2 and later)	IBM Developer Kit for Java JDBC Driver (native Driver)	X		X	V5R2 and later
DB2 UDB for iSeries (Native - V5R1 and earlier)	IBM Developer Kit for Java JDBC Driver (native Driver)	X			V5R1 and earlier
DB2 UDB for iSeries (Native XA - V5R1 and earlier)	IBM Developer Kit for Java JDBC Driver (native Driver)	X		X	V5R1 and earlier
DB2 UDB for iSeries (Toolbox)	IBM Toolbox for Java JDBC driver		X		
DB2 UDB for iSeries (Toolbox XA)	IBM Toolbox for Java JDBC driver		X	X	

When running under WebSphere Application Server on the iSeries server, the native driver is the preferred driver. The IBM Toolbox for Java JDBC driver can be used to connect to local iSeries databases; however, you can expect better performance by using the native JDBC driver for local database connections.

A.5 Administrative Console

The Administrative Console (Admin Console) in WebSphere Application Server V5.0 for iSeries is implemented as a Web application. Thus, you have to start an application server in order to use the Admin Console. Use the startServer script to start the server (see A.7.2, “Syntax and parameters for startServer script” on page 532).

To access the Admin Console:

1. Start a Web browser.
2. Open the following URL:

   `http://<iSeries host name>:<app server port>/admin`

   Here, `iSeries host name` is the host name (CFGTCP option 12) of your iSeries system. `app server port` is the port number on which the Admin Console listens for incoming requests. For more information see “Working with the Administrative Console” on page 159.

A.6 WebSphere Application Server default ports

Table 12-2 shows the default value for each port that WebSphere Application Server Version 5.0 uses for a WAS or WAS_ND default instance. All other ports for your instances are stored in the configuration file serverindex.xml. You can change the value of the other ports with the `chgwassrv` script; see 6.5.7, “Changing a WAS application server via the script” on page 128. If you create a new instance, take care of the -portblock parameter and the specific port parameters, see “Creating a new WAS instance” on page 113. To learn the syntax of the `chgwassvr` script, see A.7.4, “Syntax and parameters for chgwassvr script” on page 534.

Table 12-2 Used default ports in WebSphere Application Server Version 5.0

Default port value	Name	Description	Properties
80	external HTTP	The TCP/IP port the external HTTP server listens	
443	external HTTPS	The TCP/IP port the external HTTP server listens for SSL requests	
9080 9090 9043	internal HTTP Admin Console port Admin Console SSL port	The TCP/IP port the internal HTTP server listens The TCP/IP port for the Administrative Console The TCP/IP port for the secure connection to the Admin Console	
2809 Base 9809 ND	Name service	The TCP/IP port on which the name service listens. This port is also the RMI connector port.	BOOTSTRAP_ADDRESS
5557	JMS secure	The TCP/IP port that the internal JMS provider uses to communicate with the internal JMS provider to verify authorizations to resources when WebSphere Application Server security is enabled.	JMSSERVER_SECURITY_PORT
5558	JMS queued	The TCP/IP port that the internal JMS provider's WebSphere MQ listener uses. This listener is used by JMS connections to communicate with the internal JMS provider.	JMSSERVER_QUEUED_ADDRESS
5559	JMS direct	The TCP/IP port that the internal JMS provider uses to communicate with the internal JMS provider for JMS publish/subscribe connections when the WebSphere Topic Connection Factory resource port is set to DIRECT.	JMSSERVER_DIRECT_ADDRESS
7873 Base 7989 ND	DRS client	The TCP/IP port that your server uses for the Data Replication Service (DRS) client.	DRS_CLIENT_ADDRESS
8880 Base 8879 ND	SOAP	The TCP/IP port that your server uses for Simple Object Access Protocol (SOAP).	SOAP_CONNECTOR_ADDRESS

Default port value	Name	Description	Properties
9501 Base 9401 ND	SAS	The TCP/IP port on which the Secure Association Services (SAS) listen for inbound authentication requests.	SAS_SSL_SERVERAUTH_LISTENER_ ADDRESS
9503 Base 9403 ND	CSIV2 Server	The TCP/IP port on which the Common Secure Interoperability Version 2 (CSIV2) Service listens for inbound server authentication requests.	CSIV2_SSL_SERVERAUTH_LISTENER _ADDRESS
9502 Base 9402 ND	CSIV2 Mutual	The TCP/IP port on which the Common Secure Interoperability Version 2 (CSIV2) Service listens for inbound client authentication requests.	CSIV2_SSL_MUTUALAUTH_LISTENER _ADDRESS
9100	Object Request Broker (ORB) listener port	The TCP/IP port on which the application server Object Request Broker (ORB) listens for requests. This also the port on which the location service daemon for the node listens.	ORB_LISTENER_ADDRESS
7277	Cell discovery port	The TCP/IP port on which the node discovery service for the node agent listens.	CELL_DISCOVERY_ADDRESS
5000	Node multicast discovery port	The TCP/IP port for the multicast discovery service on which the node agent listens.	NODE_MULTICAST_DISCOVERY_ ADDRESS

A.7 QShell scripts

QShell scripts are a required element of WebSphere Application Server administration. They provide an easy and quick way to manipulate the WAS instances. This section provides information about the most useful scripts.

Most of the scripts require the *ALLOBJ authority to run.

A.7.1 Syntax and parameters for crtwasinst script

The syntax of the `crtwasinst` script is shown in Table 12-3, which contains more information and tips about the usage of the parameters:

- WebSphere Application Server

```
crtwasinst: -instance <value> [-noembeddedjms] [-exthttp <value>] [-extssl <value>]
[-os400passwords [-validationlist <value>]] [-server <value>] [-node <value>]
[-nodefaultapps] [-portblock <value>] [-inthttp <value>] [-admin <value>]
[-adminssl <value>] [-jmsqueued <value>] [-jmsdirect <value>] [-jmssecure <value>]
[-soap <value>] [-nameservice <value>] [-drsclient <value>] [-sas <value>]
[-csiv2server <value>] [-csiv2client <value>] [-verbose] [-help]
```

- WebSphere Application Server Network Deployment

```
crtwasinst -instance instance [ -portblock portblock ]
 [ -server servername ] [ -node nodename ] [ -cell cellname ]
```

```
[ -admin adminport ] [ -adminssl adminsslport ] [ -soap soapport ]
[ -orblistener orblistenerport ] [ -nameservice nameserviceport ]
[ -drsclient drsclientport ] [ -celldiscovery celldiscoveryport ]
[ -sas sasserverport ] [ -csiv2server csiv2serverauthport ]
[ -csiv2client csiv2clientauthport ] [ -verbose ] [ -help ]
```

Table 12-3 Parameters for crtwasinst script

Parameter name	Required?	Value	Creates	Notes
-instance	Yes	instance_ name	directory /QIBM/UserData/WebAS5/Base/instance_name or /QIBM/UserData/WebAS5/ND/instance_name	
-portblock	Optional, but always recommended	port_number Specifies the first number of a block of port numbers that your instance uses. Define here the first port in a group of unused ports on your iSeries server. Fore all of the possible ports you can also use the available specific port parameters in the crtwasinst script. When you create a new instance, ports are assigned based on the following ordered conditions: **Specific port parameters:** If you specify values for specific port parameters, the script uses those values. **The -portblock parameter:** Services for which you have not specified a specific port number are assigned to ports in sequence starting with the value of the -portblock parameter.	The port numbers for the WAS services are stored in the serverindex.xml file. You can change the value by using the chgwassrv script.	You can use the Work with TCP/IP Network Status (NETSTAT *CNN) command to display a list of port numbers that are currently being used. The -portblock parameter does not affect the HTTP ports. If you specify the -portblock parameter, but not the specific HTTP ports parameters, your instance uses the default values for the HTTP ports.

Parameter name	Required?	Value	Creates	Notes
-exthttp	Optional	port number Specifies the number of the TCP/IP port where the external HTTP server listens. The default value is 80.	You can change the external HTTP port via the administrative console, see Chapter , “Updating Host alias table” on page 162.	The -portblock parameter does not affect the external HTTP port. If you specify the -portblock parameter, but not the -exthttp parameter, your instance uses the default value 80 for the external HTTP port.
-extssl	Optional	port number The value extsslport specifies the number of the TCP/IP port where the external SSL-enabled HTTP server listens. The default value is 443.	You can change the external SSL HTTP port via the administrative console, see Chapter , “Updating Host alias table” on page 162.	The -portblock parameter does not affect the external SSL port. If you specify the -portblock parameter, but not the -extssl parameter, your instance uses the default value 443 for the external SSL port.
-inthttp	Optional	port number The value inthttpport specifies the port number on which the internal WAS HTTP server listens. If neither the -portblock parameter nor the -inthttp parameter is specified, the default value is 9080.	You can change the internal HTTP port via the administrative console, see Chapter , “Updating Host alias table” on page 162.	
-admin	Optional	port number The value adminport specifies the port number to use for the administrative console. If neither the -portblock parameter nor the -admin parameter is specified, the default value is 9090.	You can change the adminport via the administrative console, see Chapter , “Updating Host alias table” on page 162.	
-adminssl	Optional	port number The value adminportssl specifies the port number to use for the secure communications with administrative console. If neither the -portblock parameter nor the -adminssl parameter is specified, the default value is 9043.	You can change the adminport via the administrative console, see Chapter , “Updating Host alias table” on page 162.	

Parameter name	Required?	Value	Creates	Notes
-jmsqueued	Optional	port number Only Base application server. The value jmsqueuedport specifies the port number that the internal JMS provider's WebSphere MQ listener uses. This listener is used by JMS connections to communicate with the internal JMS provider. If neither the -portblock parameter nor the -jmsqueued parameter is specified, the default value is 5558.	The port numbers for the WAS services are stored in the serverindex.xml file. You can change the value by using the chgwassrv script.	
-jmsdirect	Optional	port number Only Base application server. The value jmsdirectport specifies the port that the internal JMS provider uses to communicate with the internal JMS provider for JMS publish/subscribe connections when the WebSphere Topic Connection Factory resource port is set to DIRECT. If neither the -portblock parameter nor the -jmsdirect parameter is specified, the default value is 5559.	The port numbers for the WAS services are stored in the serverindex.xml file. You can change the value by using the chgwassrv script.	

Parameter name	Required?	Value	Creates	Notes
-jmssecure	Optional	port number Only Base application server. The value jmssecureport specifies the port that the internal JMS provider uses to communicate with the internal JMS provider to verify authorizations to resources when WebSphere Application Server security is enabled. If neither the -portblock parameter nor the -jmssecure parameter is specified, the default value is 5557.	The port numbers for the WAS services are stored in the serverindex.xml file. You can change the value by using the chgwassrv script.	
-soap	Optional	port number The value soapport specifies the port number to use for Simple Object Access Protocol (SOAP). If neither the -portblock parameter nor the -soap parameter is specified, the script assigns the default value. The default value for a WebSphere Application Server instance is 8880. The default value for a WebSphere Application Server Network Deployment instance is 8879.	The port numbers for the WAS services are stored in the serverindex.xml file. You can change the value by using the chgwassrv script.	

Parameter name	Required?	Value	Creates	Notes
-nameservice	Optional	port number The value nameserviceport specifies the port number to use for name service (or RMI connector) port. If neither the -portblock parameter nor the -nameservice parameter is specified, the script assigns the default value. The default value for WebSphere Application Server is 2809. The default value for WebSphere Application Server Network Deployment is 9809	The port numbers for the WAS services are stored in the serverindex.xml file. You can change the value by using the chgwassrv script.	
-drsclient	Optional	port number The value drsclientport specifies the port number to use for the Data Replication Service (DRS) client. If neither the -portblock parameter nor the -drsclient parameter is specified, the script assigns the default value. The default value for WebSphere Application Server is 7873. The default value for WebSphere Application Server Network Deployment is 7989.	The port numbers for the WAS services are stored in the serverindex.xml file. You can change the value by using the chgwassrv script.	

Parameter name	Required?	Value	Creates	Notes
-sas	Optional	port number The value sasserverport specifies the port on which the Secure Association Services (SAS) listen for inbound authentication requests. For WebSphere Application Server, the default value is 9501. For WebSphere Application Server Network Deployment, the default value is 9401.	The port numbers for the WAS services are stored in the serverindex.xml file. For information on changing this port with the administrative console, see End point settings.	It is recommended that you specify this parameter. If you do not specify this parameter, the application server selects a port at runtime. However, if a client application is connected to the application server and the application server restarts, the server could select a different port number, and the client application would be unable to connect to the server.
-csiv2server	Optional	port number The value csiv2serverauthport specifies the port on which the Common Secure Interoperability Version 2 (CSIV2) Service listens for inbound server authentication requests. For WebSphere Application Server, the default value is 9503. For WebSphere Application Server Network Deployment, the default value is 9403.	The port numbers for the WAS services are stored in the serverindex.xml file. You can change the value by using the chgwassrv script.	It is recommended that you specify this parameter. If you do not specify this parameter, the application server selects a port at runtime. However, if a client application is connected to the application server and the application server restarts, the server could select a different port number, and the client application would be unable to connect to the server.
-csiv2client	Optional	port number The value csiv2clientauthport specifies the port on which the Common Secure Interoperability Version 2 (CSIV2) Service listens for inbound client authentication requests. For WebSphere Application Server, the default value is 9502. For WebSphere Application Server Network Deployment, the default value is 9402.	The port numbers for the WAS services are stored in the serverindex.xml file. You can change the value by using the chgwassrv script.	It is recommended that you specify this parameter. If you do not specify this parameter, the application server selects a port at runtime. However, if a client application is connected to the application server and the application server restarts, the server could select a different port number, and the client application would be unable to connect to the server.

Parameter name	Required?	Value	Creates	Notes
-orblistener	Optional	port number Only Network Deployment. The value orblistenerport specifies the port number to use for the orb listener port. If neither the -portblock or -orblistener parameter is specified, the default is 9100.	The port numbers for the WAS services are stored in the serverindex.xml file. You can change the value by using the chgwassrv script.	
-celldiscovery	Optional	port number Only Network Deployment. The value celldiscoveryport specifies the port on which the Network Deployment instance listens for a node attempting to find the cell to which the node belongs. If neither the -portblock parameter nor the -celldiscovery parameter is specified, the default value is 7277.	The port numbers for the WAS services are stored in the serverindex.xml file. You can change the value by using the chgwassrv script.	
-noembeddedjms	Optional	Specify this parameter if you do not want to create an embedded Java Message Service (JMS) provider for your instance.	If you do not specify this parameter an embedded JMS provider is created, when you installed WebSphere MQ for iSeries and WebSphere MQ classes for Java and JMS V5.3 for iSeries	
-server	Optional	server_name If this value is not specified, the application server name is the same as the instance name.	specifies the name of the application server that runs in your instance	
-nodefaultapps	Optional	Specifies that the default applications should not be installed in the instance. The default applications provide simple examples that you can use to verify that your application server is functioning properly.		

Parameter name	Required?	Value	Creates	Notes
-node	Optional	node_name The default node name is hostname_instance, where hostname is the host name of the iSeries server and instance is the value specified for the -instance parameter. For a WebSphere Application Server Network Deployment instance, the default node name is instanceManager, where instance is the value specified for the -instance parameter If this parameter is not specified, the script assigns the default value		
-os400passwords	Optional	If you specify this parameter, the script sets the os400.security.password.encoding.algorithm property in the application server's server.xml file to OS400. The default encoding algorithm is XOR.		
-validationlist	Optional	listpath The value listpath specifies the path to the validation list or lists that are used for the OS400 password encoding algorithm. It sets the os400.security.password.validation.list.object property in the application server's server.xml file. This parameter only applies if the -os400passwords parameter is included. The default value is /QSYS.LIB/QUSRSYS.LIB/EJSADMIN.VLDL.		

Parameter name	Required?	Value	Creates	Notes
-cell	Optional	Only for Network Deployment. The value cellname specifies the name of the cell for the instance. If this parameter is not specified, the cell name is instanceNetwork where instance is the value specified for the -instance parameter.		
-verbose	Optional	Turns on verbose messages, which can be helpful if you need to debug the script.		
-help	Optional	Displays the help message. If you specify this parameter, the script ignores all other parameters.		

A.7.2 Syntax and parameters for startServer script

The syntax of the `startServer` script is shown in Table 12-4, which contains more information and tips about the usage of the parameters:

```
startServer [ server ] [ -instance instance ]
 [ -nowait ] [ -timeout seconds ] [ -help | -? ]
```

Here, `server` specifies the name of the server that you want to start. If you omit this parameter, the script will attempt to start `server1` of the `default` instance.

Table 12-4 Parameters of the startServer script

Parameter name	Required?	Value	Notes
-instance	No	instance_name	
-nowait	Optional	none	Tells the startServer script not to wait for successful initialization of the launched server process.
-timeout	Optional	seconds	Specifies the time to wait before server initialization times out and returns an error.
-help	Optional	none	Displays the help message. If you specify this parameter, the script ignores all other parameters.

A.7.3 Syntax and parameters for stopServer script

The syntax of the `stopServer` script is shown in Table 12-5, which contains more information about the usage of the parameters:

```
stopServer  server [ -instance instance ] [ -nowait ] [ -quiet ]
 [ -logfile filename ] [ -replacelog ] [ -trace ] [ -timeout seconds ]
 [ -statusport statusportnumber ] [ -port portnumber ]
 [ -username username ] [ -password password ] [ -conntype SOAP | RMI ]
 [ -help | -? ]
```

Here, `server` is the name of the application server you want to stop. This argument is required.

Table 12-5 Parameter of the stopServer script

Parameter name	Required?	Value	Notes
-instance	No	instance_name	
-nowait	Optional	none	Tells stopServer script not to wait for successful shutdown of server process.
-quiet	Optional	none	Suppresses the progress information that the stopServer script prints in normal mode.
-logfile	Optional	filename	Specifies the location of the log file to which information is written.
-replacelog	Optional	none	Replaces the log file instead of appending to the current log.
-trace	Optional	none	Generates trace information into a file for debugging purposes.
-timeout	Optional	seconds	Specifies the time to wait for server shutdown before timing out and returning an error.
-statusport	Optional	statusportnumber	Allows an administrator to set the port number for server status callback.
-port	Optional	portnumber	Specifies the server Java Management Extensions (JMX) port to use explicitly, so that you can avoid reading the configuration files to obtain the information.
-user	Optional	username	Specifies the user name for authentication if security is enabled in the server.
-password	Optional	password	Specifies the password for authentication if security is enabled in the server.
-conntype	Optional	type	Specifies the Java Management Extensions (JMX) connector type to use for connecting to the deployment manager. Valid types are Simple Object Access Protocol (SOAP), or Remote Method Invocation (RMI).
-help	Optional	none	Displays the help message.

A.7.4 Syntax and parameters for chgwassvr script

The syntax of the `chgwassvr` script is shown in Table 12-6, which contains more information about the usage of the parameters:

WebSphere Application Server:

```
chgwassvr -server servername [ -instance instance ]
 [ -embeddedjms yes|no ] [ -portblock portblock ] [ -inthttp inthttpport ]
 [ -inthttpssl inthttpsslport ] [ -admin adminport ]
 [ -adminssl adminsslport ] [ -jmsqueued jmsqueuedport ]
 [ -jmsdirect jmsdirectport ] [ -jmssecure jmssecureport ]
 [ -soap soapport ] [ -nameservice nameserviceport ]
 [ -drsclient drsclientport ] [ -sas sasserverport ]
 [ -csiv2server csiv2serverauthport ] [ -csiv2client csiv2clientauthport ]
 [ -verbose ] [ -help ]
```

WebSphere Application Server Network Deployment

```
chgwassvr -server servername [ -instance instance ] [ -node nodename ]
 [ -portblock portblock ] [ -inthttp inthttpport ]
 [ -inthttpssl inthttpsslport ] [ -admin adminport ]
 [ -adminssl adminsslport ] [ -jmsqueued jmsqueuedport ]
 [ -jmsdirect jmsdirectport ] [ -jmssecure jmssecureport ]
 [ -orblistener orblistenerport ] [ -soap soapport ]
 [ -orblistener orblistenerport ] [ -nameservice nameserviceport ]
 [ -drsclient drsclientport ] [ -celldiscovery celldiscoveryport ]
 [ -sas sasserverport ] [ -csiv2server csiv2serverauthport ]
 [ -csiv2client csiv2clientauthport ] [ -nodediscovery nodediscport ]
 [ -nodemulti nodemultdiscport ] [ -verbose ] [ -help ]
```

Table 12-6 Parameters of the chgwassvr script

Parameter name	Required?	Value	Notes
-instance	No	instance_name	The value instance specifies the name of the instance that contains the application server you want to change. The default value is `default`.
-server	Yes	servername	The value servername specifies the name of the application server to change.
-node	Only Network Deployment optional	nodename	The value nodename specifies the node that hosts the application server that you want to change. The default value is the Network Deployment managing instance.
-embeddedjms	Optional	yes or no	The value specifies whether the server should contain an embedded JMS configuration. If this parameter is not specified, the JMS configuration for the application server is unchanged. If the value is yes, an embedded JMS configuration is created if the server does not have one. If the value is no, the JMS configuration is deleted from the server if it has one.

Parameter name	Required?	Value	Notes
-portblock	Optional	portblock	The value portblock specifies the first number of a block of port numbers that your instance uses. If you specify this parameter, the script changes all of the port numbers for your application server. If you do not specify this parameter, the port numbers for your application server are not changed, unless you specify a port parameter to change (see port parameters below). When you change an application server's properties, ports are assigned based on the following ordered conditions: Specific port parameters If you specify values for specific port parameters, the script uses those values. The -portblock parameter Services for which you have not specified a specific port number are assigned ports sequentially starting with the value of the -portblock parameter. If a script encounters a port that is in use, it skips that port number and continues with the next unused port. You can use the Work with TCP/IP Network Status (NETSTAT *CNN) command to display a list of port numbers that are currently being used.
-inthttp	Optional	port number	The value inthttpport specifies the port number of the WAS internal HTTP port. If neither the -portblock parameter nor the -inthttp parameter is specified, this port is not changed. See the Notes on the -portblock parameter for more information.
-admin	Optional	port number	The value adminport specifies the port number to use for the administrative console. If neither the -portblock parameter nor the -admin parameter is specified, this port is not changed. See the Notes on the -portblock parameter for more information.
-adminssl	Optional	port number	The value adminportssl specifies the port number to use for the secure communications with administrative console. If neither the -portblock parameter nor the -adminssl parameter is specified, this port is not changed. See the Notes on the -portblock parameter for more information.

Parameter name	Required?	Value	Notes
-jmsqueued	Optional	port number	The value jmsqueuedport specifies the port number that the internal JMS provider's WebSphere MQ listener uses. This listener is used by JMS connections to communicate with the internal JMS provider. If neither the -portblock parameter nor the -jmsqueued parameter is specified, this port is not changed. See the Notes on the -portblock parameter for more information.
-jmsdirect	Optional	port number	The value jmsdirectport specifies the port that the internal JMS provider uses to communicate with the internal JMS provider for JMS publish/subscribe connections when the WebSphere Topic Connection Factory resource port is set to DIRECT. If neither the -portblock parameter nor the -jmsdirect parameter is specified, this port is not changed. See the Notes on the -portblock parameter for more information.
-jmssecure	Optional	port number	The value jmssecureport specifies the port that the internal JMS provider uses to communicate with the internal JMS provider to verify authorizations to resources when WebSphere Application Server security is enabled. If neither the -portblock parameter nor the -jmssecure parameter is specified, this port is not changed. See the Notes on the -portblock parameter for more information.
-soap	Optional	port number	The value soapport specifies the port number to use for Simple Object Access Protocol (SOAP). If neither the -portblock parameter nor the -soap parameter is specified, this port is not changed. See the Notes on the -portblock parameter for more information.
-orblistener	Only Network Deployment optional	port number	The value orblistenerport specifies the port number to use for the orb listener port. If neither the -portblock or -orblistener parameter is specified, this port is not changed. See the Notes on the -portblock parameter for more information.

Parameter name	Required?	Value	Notes
-nameservice	Optional	port number	The value nameserviceport specifies the port number to use for name service (or RMI connector) port. If neither the -portblock parameter nor the -nameservice parameter is specified, this port is not changed. See the Notes on the -portblock parameter for more information.
-drsclient	Optional	port number	The value drsclientport specifies the port number to use for the Data Replication Service (DRS) client. If neither the -portblock parameter nor the -drsclient parameter is specified, this port is not changed. See the Notes on the -portblock parameter for more information.
-celldiscovery	Only Network Deployment optional	port number	The value celldiscoveryport specifies the port on which the Network Deployment instance listens for a node attempting to find the cell to which the node belongs. If neither the -portblock parameter nor the -celldiscovery parameter is specified, this port is not changed. See the Notes on the -portblock parameter for more information.
-sas	Optional	port number	The value sasserverport specifies the port on which the Secure Association Services (SAS) listen for inbound authentication requests. If the -sas parameter is not specified, this port is not changed.
-csiv2server	Optional	port number	The value csiv2serverauthport specifies the port on which the Common Secure Interoperability Version 2 (CSIV2) Service listens for inbound server authentication requests. If the -csiv2server parameter is not specified, this port is not changed.
-csiv2client	Optional	port number	The value csiv2clientauthport specifies the port on which the Common Secure Interoperability Version 2 (CSIV2) Service listens for inbound client authentication requests. If the -csiv2client parameter is not specified, this port is not changed.
-nodediscovery	Optional	port number	The value nodediscport specifies the port on which the node agent's node discovery service listens.

Parameter name	Required?	Value	Notes
-nodemulti	Optional	port number	The value nodemultidiscport specifies the port on which the node agent's multicast discovery service listens.
-verbose	Optional		This optional parameter turns on verbose messages, which can be helpful if you need to debug the script.
-help	Optional	none	Displays the help message.

A.7.5 Syntax and parameters for dltwasinst script

The syntax of the `dltwasinst` script is shown in Table 12-7, which contains more information about the usage of the parameters:

```
dltwasinst -instance instance [ -verbose ] [ -help ]
```

Table 12-7 Parameters of the dltwasinst script

Parameter name	Required?	Value	Notes
-instance	Yes	instance_name	The value instance specifies the name of the instance that you want to delete.
-verbose	Optional	none	Turns on verbose messages, which can be helpful if you need to debug the script
-help	Optional	none	Displays the help message.

A.7.6 Syntax and parameters of the startManager script

The syntax of the `startManager` script is:

```
startManager [ server ] [ -instance instance ]
 [ -nowait ] [ -timeout seconds ] [ -help | -? ]
```

`server` is an optional parameter that specifies the name of the deployment manager that you want to start.

If `server` is not specified and -instance is not specified or has a value of default, the script will attempt to start a default server - `dmgr`.

If `server` is not specified and a non-default instance is specified for the -instance parameter, the deployment manager name defaults to the value specified for the -instance parameter.

The server name value is case sensitive.

The parameters for the `startManager` script are shown in Table 12-8.

Table 12-8 Parameters of the startManager script

Parameter name	Required?	Value	Notes
none	Optional	server name	This is an optional parameter specifiyng the name of the deployment manager that you want to start. If neither server nor the -instance parameter is specified, the default value for server is dmgr. If server is not specified and the value specified for -instance parameter is default, the value for server is dmgr. If server is not specified and a non-default instance is specified for the -instance parameter, the deployment manager name defaults to the value specified for the -instance parameter. This value is case sensitive.
-instance	Optional	instance_name	The value instance specifies the name of the Network Deployment instance associated with the deployment manager that you want to start. The default value for this parameter is default
-nowait	Optional	none	If you specify this parameter, the script returns control to the user without waiting for successful initialization of the server. The default is to wait for successful initialization.
-timeout	Optional	none	The value seconds specifies the amount of time in seconds to wait for successful initialization of the server. The script returns control to the user at the end of the timeout value. The default is to wait until the server initialization is complete.
-help or ?	Optional	none	Prints the usage statement for the script.

A.7.7 Syntax and parameters for the stopManager script

The syntax of the stopManager script is:

```
stopManager  server [ -instance instance ] [ -nowait ] [ -quiet ]
 [ -logfile filename ] [ -replacelog ] [ -trace ] [ -timeout seconds ]
 [ -statusport statusportnumber ] [ -port portnumber ]
 [ -username username ] [ -password password ] [ -conntype SOAP | RMI ]
 [ -help | -? ]
```

The parameter `server` is required. The value of server specifies the name of the deployment manager that you want to stop. This value is case sensitive.

The parameters for the `stopManager` script are shown in Table 12-9.

Table 12-9 Parameters of the stopManager script

Parameter name	Required?	Value	Notes
none	Required	server name	The value server specifies the name of the deployment manager that you want to stop. This value is case sensitive.
-instance	Optional	instance_name	The value instance specifies the name of the instance associated with the deployment manager that you want to stop. The default value is default.
-nowait	Optional	none	If you specify this parameter, the script returns control to the user without waiting for the deployment manager to stop successfully. The default is to wait for the deployment manager to stop successfully.
-quiet	Optional	none	If you specify this parameter, the script does not display informational messages. The default is to display informational messages while the script runs.
-logfile	Optional	location and file name	The value filename specifies the location and name of the log file for the script. The default value is /QIBM/UserData/WebAS5/ND/instance/logs/server/stopServer.log where instance is the name of the instance associated with the deployment manager that you want to stop, and server is the name of the deployment manager that you want to stop.
-replacelog	Optional	none	If you specify this parameter, the script replaces the log file if it exists. By default the script appends to the log file if it exists.
--trace	Optional	none	If you specify this parameter, the script outputs additional trace information to the log file for the script. You should only specify this parameter if errors occur when you try to stop a deployment manager. The default is to not log additional trace information
-timeout	Optional	none	The value seconds specifies the amount of time in seconds to wait for the deployment manager to stop before returning control to the caller. The default is to wait until the deployment manager has stopped successfully.
-statusport	Optional	port number	The value statusportnumber specifies the port on which to listen for the status of the deployment manager while it is stopping. The default is to use the next available port.

Parameter name	Required?	Value	Notes
-port	Optional	port number	The value portnumber specifies the SOAP or RMI port for the deployment manager. If you specify this parameter, the stopManager script sends the stop command directly to deployment manager. If you specify the RMI port value for this parameter, you must also specify the -conntype parameter. By default, the script reads the configuration files to obtain the information that is necessary to stop the deployment manager
-conntype	Optional	SOAP or RMI	If you specify the -port parameter, the -conntype parameter specifies the connector type to use. Valid values are SOAP or RMI. The default value is SOAP
-username	Required if security is enabled for the deployment manager	user name	The value username specifies the user name for authentication.
-password	Required if security is enabled for the deployment manager	password	The value password specifies the password for authentication.
-help or ?	Optional	none	Prints the usage statement for the script.

A.7.8 The syntax of the addNode script

The syntax for the `addNode` script is:

```
addNode cell_host [ cell_port ] [ -instance instance ]
 [ -nowait ] [ -quiet ] [ -logfile logfilename ] [ -replacelog ] [ -trace ]
 [ -noagent ] [ -username username ] [ -password password ]
 [-conntype conntype ] [ -includeapps ] [ -startingport startportnumber ]
 [ -help | -? ]
```

Parameters for the `addNode` script are shown in Table 12-10.

Table 12-10 Parameters of the addNode script

Parameter name	Required?	Value	Notes
none	Yes	cell_host name	The value cell_host specifies the name of the machine that hosts the deployment manager (Network Deployment instance) for the Network Deployment cell.
none	Yes	SOAP or RMI cell_port	The value cell_port specifies the SOAP or RMI port on which the deployment manager for the Network Deployment cell is listening. By default, the script uses the SOAP connector type. If you specify a port other than the SOAP port for your deployment manager, you must also specify the -conntype parameter. Port 8879 is the default SOAP port for the default instance of Network Deployment. The deployment manager must be started and actively listening on the port specified for this parameter.
-nowait	Optional	none	If you specify this parameter, the script returns control to the user without waiting for the script to finish adding the node. The default is to wait for the script to complete.
-quiet	Optional	none	If you specify this parameter, the script does not display informational messages. The default is to display informational messages while the script runs.
-logfile	Optional	directory and filename	The value filename specifies the location and name of the log file for the script. The default value is /QIBM/UserData/WebAS5/Base/instance/logs/addNode.log where instance is the name of the instance that contains to which the server for which you want to display status.
-replacelog	Optional	none	If you specify this parameter, the script replaces the log file if it exists. By default the script appends to the log file if it exists.
-trace	Optional	none	If you specify this parameter, the script outputs additional trace information to the log file for the script. You should only specify this parameter if errors occur when you try to add a node. The default is to not log additional trace information.

Parameter name	Required?	Value	Notes
-noagent	Optional	none	If you specify this parameter, the script does not launch the node agent process automatically after the node is added to the cell. The default is to start the node agent process automatically as part of the addNode script.
-username	Required if security is enabled for the server	username	The value username specifies the user name for authentication.
-password	Required if security is enabled for the server	password	The value password specifies the password for authentication.
-conntype	Optional	SOAP or RMI	If you specify the -port parameter, the -conntype parameter specifies the connector type to use. Valid values are SOAP or RMI. The default value is SOAP.
-includeapps	Optional	none	By default, when the script adds the node to the Network Deployment cell, it does not include the applications that are installed on the application server. If you specify this parameter, the script installs the applications into the Network Deployment cell when it adds the node. If the application already exists in the cell, the script prints warning and the application is not installed into the cell.
-startingport	Recommended	startportnumber	The value startportnumber specifies the first of a block of port numbers that the script uses for the node agent and JMS server ports that it creates. This parameter allows you to control which ports are defined for these servers. If you do not specify this parameter, the script assigns the default port numbers to the node agent and JMS server. If you are using multiple nodes, it is recommended that you specify this parameter.
-help or ?	Optional	none	Prints the usage statement for the script.

A.7.9 The syntax of the startNode script

The syntax for the `startNode` script is:

```
startNode [ -instance instance ] [ -nowait ]
[ -timeout seconds ] [ -help | ? ]
```

The parameters for the `startNode` script are shown in Table 12-11.

Table 12-11 Parameters of the startNode script

Parameter name	Required?	Value	Notes
-instance	Optional but recommended	instance_name	The value instance specifies the name of the server for which you want to start the node agent. The default value for this parameter is default.
-nowait	Optional	none	If you specify this parameter, the script returns control to the user without waiting for successful initialization of the server. The default is to wait for successful initialization.
-timeout	Optional	seconds	The value seconds specifies the amount of time in seconds to wait for successful initialization of the server. The script returns control to the user at the end of the timeout value. The default is to wait until the server initialization is complete.
-help or ?	Optional	none	Prints the usage statement for the script

A.7.10 The syntax of the stopNode script

The syntax for the `stopNode` script is:

```
stopNode [ -instance instance ] [ -nowait ] [ -quiet ] [ -logfile filename ]
 [ -replacelog ] [ -trace ] [ -timeout seconds ] [ -statusport statusportnumber ]
 [ -port portnumber ] [ -username username ] [ -password password ]
 [ -conntype type ] [ -help | -? ]
```

The parameters for the `stopNode` script are shown in Table 12-11.

Table 12-12 Parameters of the stopNode script

Parameter name	Required?	Value	Notes
-instance	Optional but recommended	instance_name	The value instance specifies the name of the server for which you want to start the node agent. The default value for this parameter is default.
-nowait	Optional	none	If you specify this parameter, the script returns control to the user without waiting for successful initialization of the server. The default is to wait for successful initialization.
-quiet	Optional	none	If you specify this parameter, the script does not display informational messages. The default is to display informational messages while the script runs.

Parameter name	Required?	Value	Notes
-logfile	Optional	file name	The value filename specifies the location and name of the log file for the script. The default value is /QIBM/UserData/WebAS5/Base/<instance>/logs/nodeagent/stopServer.log where instance is the name of the instance for which you want to stop the node agent.
-replacelog	Optional	none	If you specify this parameter, the script replaces the log file if it exists. By default the script appends to the log file if it exists.
-trace	Optional	none	If you specify this parameter, the script outputs additional trace information to the log file for the script. You should only specify this parameter if errors occur when you try to stop a node agent. The default is to not log additional trace information.
-statusport	Optional	statusportnumber	The value statusportnumber specifies the port on which to listen for the status of the node agent while it is stopping. The default is to use the next available port.
-port	Optional	portnumber	The value portnumber specifies the SOAP or RMI port for the node agent. If you specify this parameter, the stopNode script sends the stop command directly to node agent. If you specify the RMI port value for this parameter, you must also specify the -conntype parameter. By default, the script reads the configuration files to obtain the information that is necessary to stop the node agent.
-conntype	Optional	SOAP or RMI	If you specify the -port parameter, the -conntype parameter specifies the connector type to use. Valid values are SOAP or RMI. The default value is SOAP.
-username	Required if the security is enabled	username	The value username specifies the user name for authentication.
-password	Required if the security is enabled	password	The value password specifies the password for authentication.

Parameter name	Required?	Value	Notes
-timeout	Optional	seconds	The value seconds specifies the amount of time in seconds to wait for successful initialization of the server. The script returns control to the user at the end of the timeout value. The default is to wait until the server initialization is complete.
-help or ?	Optional	none	Prints the usage statement for the script

A.7.11 The syntax of the GenPluginCfg script

The syntax for the `GenPluginCfg` script is:

```
GenPluginCfg [ -instance instance ] [ -config.root cfgdirectory ]
 [ -cell.name cellname ] [ -node.name nodename ]
 [ -server.name servername ] [ -output.file.name filename ]
 [ -destination.root destinationdir  ] [ -destination.operating.system destOS ]
 [ -debug yes | no ]
```

The parameters for the `GenPluginCfg` script are shown in Table 12-13.

Table 12-13 Parameters of the GenPluginCfg script

Parameter name	Required?	Value	Notes
-instance	Optional but recommended	instance	The value instance specifies the name of your application server instance. The default value is default.
-config.root	Optional	cfgdirectory	The value cfgdirectory specifies the configuration directory that contains the plugin-cfg.xml file. The default value is the /QIBM/UserData/WebAS5/edition/instance/config/cells directory, where edition is Base for WebSphere Application Server and ND for WebSphere Application Server Network Deployment, and instance is the name of your instance.
-cell.name	Optional	cellname	The value cellname specifies the name of the cell in which your application server resides. The default is the value specified for the WAS_CELL environment variable found in file /QIBM/UserData/WebAS5/edition/instance/bin/setupCmdLine, where edition is Base for WebSphere Application Server and ND for WebSphere Application Server Network Deployment, and instance is the name of your instance. This value is case sensitive.

Parameter name	Required?	Value	Notes
-node.name	Optional	nodename	The value nodename specifies the name of the node in which your application server resides. The default is the value specified for the WAS_NODE environment variable found in file /QIBM/UserData/WebAS5/edition/instance/bin/setupCmdLine, where edition is Base for WebSphere Application Server and ND for WebSphere Application Server Network Deployment, and instance is the name of your instance. This value is case sensitive.
-server.name	Optional	servername	The value servername specifies the name of the application server for which you want to regenerate the plugin-cfg.xml file. In a Network Deployment environment, specify this option to regenerate the configuration for a single application server in your cell. If you do not specify this parameter, the script regenerates the plugin configurations for all of the servers in the node. This value is case sensitive.
-output.file.name	Optional	filename	The value filename specifies the name of the file for the new plugin configuration. The default value is the plugin-cfg.xml file in the /QIBM/UserData/WebAS5/edition/instance/config/cells directory, where edition is Base for WebSphere Application Server and ND for WebSphere Application Server Network Deployment, and instance is the name of your instance.
-destination.root	Optional	destinationdir	The value destinationdir specifies the destination root directory that the generated output file is copied to. This directory is used as a reference for generating the paths of certain files within the generated output file, such as the log file. The default value is the /QIBM/UserData/WebAS5/edition/instance directory, where edition is Base for WebSphere Application Server and ND for WebSphere Application Server Network Deployment.

Parameter name	Required?	Value	Notes
-destination.operating.system	Optional	destOS	The value destOS specifies the operating system of the machine that the generated output file is copied to. This parameter determines the file separator that is used to form file paths in the output file. Valid values are windows or unix. If you specify windows, the file separator is a backward slash (\); if you specify unix, the file separator is a forward slash (/). The default value is unix. When the destination operating system is iSeries, use the default value.
-debug	Optional	yes or no	If you specify yes, the script generates exception trace information. If the script throws exceptions, run it with this parameter to view the trace information. The default value is no.

A.8 PTF information

To get the latest information about the required or recommended PTFs, point your Web browser to the following URL:

http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/services/service.htm

A.8.1 Group PTF numbers

Table 12-14 summarizes information about WebSphere Application Server group PTFs.

Table 12-14 Group PTF numbers

Product	V5R1	V5R2
WebSphere Application Server - base	SF99243	SF99245
WebSphere Application Server - ND	SF99244	SF99246
Database (included in the WAS group PTF)	SF99501	SF99502
Java (included in the WAS group PTF)	SF99069	SF99169
HTTP Server (included in the WAS group PTF)	SF99156	SF99098

A.9 Other Web sites

Web sites that pertain to specific subject areas are listed in this appendix in the area for each subject. Here we list some Web sites of a more general nature:

- *Using AS/400 Database Monitor and Visual Explain to Identify and Tune SQL Queries*, REDP0502:

  http://www.redbooks.ibm.com/redpapers/pdfs/redp0502.pdf

- *iSeries Performance Capabilities Reference V5R2, June 2002 Edition*, SC41-0607:

  http://publib.boulder.ibm.com/iseries/v5r2/ic2924/books/as4ppcp5.pdf

- iSeries information center:

  http://publib.boulder.ibm.com/pubs/html/as400/infocenter.html

- iSeries redbooks:

  http://publib-b.boulder.ibm.com/redbooks.nsf/portals/AS400Redbooks

- iSeries Security White Paper, *Security is fundamental to the success of doing e-business*:

  http://www.ibm.com/security/library/wp_secfund.shtml

B

Additional material

This redbook refers to additional material that can be downloaded from the Internet as described below.

Locating the Web material

The Web material associated with this redbook is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

ftp://www.redbooks.ibm.com/redbooks/SG246588

Alternatively, you can go to the IBM Redbooks Web site at:

ibm.com/redbooks

Select the **Additional materials** and click **Click here for the Redbooks FTP Server**. Open the directory that corresponds with the redbook form number, SG246588.

Using the Web material

The additional Web material that accompanies this redbook includes the following files:

File name	*Description*
sw6588.zip	This file contains zipped code samples

System requirements for downloading the Web material

The following system configuration is recommended:

Hard disk space:	20MB minimum
Operating System:	Windows NT/2000
Processor:	Pentium III or higher
Memory:	512 MB for using the workstation tools

How to use the Web material

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material zip file into this folder. You should see 2 subfolders:

- MyBankCMP application
- MDBSamples application

Each of the folders contains a sample application that has been used in the book.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 554.

- *Migrating to WebSphere V5.0: An End-to-End Migration Guide,* SG24-6910-01
- *HTTP Server (powered by Apache): An Integrated Solution for IBM eServer iSeries Servers,* SG24-6716-00
- *IBM WebSphere Application Server V5.0 System Management and Configuration: WebSphere Handbook Series,* SG24-6195-00
- *WebSphere Studio Application Developer Version 5 Programming Guide,* SG24-6957-00

Referenced Web sites

These Web sites are also relevant as further information sources:

- WebSphere Application Server for iSeries Information Center

  `http://publib.boulder.ibm.com/iseries/v1r1m0/websphere/ic2924/index.htm?info/rzaiz/50/was.htm`
- WebSphere Application Server InfoCenter

  `http://publib7b.boulder.ibm.com/webapp/wasinfo1/index.jsp?deployment=NetworkDeployment`
- The Java Management Extensions (JMX) home page

  `http://java.sun.com/products/JavaManagement/`
- Sample Scripts for WebSphere Application Server Version 5

  `http://www7b.boulder.ibm.com/wsdd/library/samples/SampleScripts.html`
- Domino help documentation

  `http://www.lotus.com/ldd/doc/domino_notes/5.0/as400/as400hlp.nsf`
- IBM eserver iSeries support, software knowledge base

  `http://www-912.ibm.com/s_dir/slkbase.nsf/slkbase`
- IBM JDK downloads

  `http://www.ibm.com/developerworks/java/jdk/`
- IBM Toolbox for Java downloads Web page

  `http://www.ibm.com/servers/eserver/iseries/toolbox/downloads.htm`
- PTDV Web page

  `http://alphaworks.ibm.com/tech/ptdv`
- Performance Management resource library

  `http://www.ibm.com/servers/eserver/iseries/perfmgmt/resource.htm`

How to get IBM Redbooks

You can order hardcopy Redbooks, as well as view, download, or search for Redbooks at the following Web site:

ibm.com/redbooks

You can also download additional materials (code samples or diskette/CD-ROM images) from that site.

IBM Redbooks collections

Redbooks are also available on CD-ROMs. Click the CD-ROMs button on the Redbooks Web site for information about all the CD-ROMs offered, as well as updates and formats.

Index

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

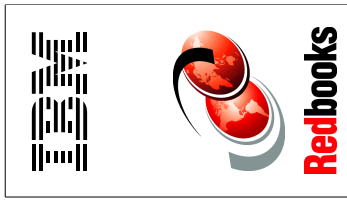
T

U

V

W

X



WebSphere Application Server V5 for iSeries: Installation, Configuration, and Administration

(1.0” spine)
0.875”<->1.498”
460 <-> 788 pages

WebSphere Application Server V5 for iSeries:

Installation, Configuration, and Administration

Redbooks

Covers both Base and Network Deployment editions

All-in-one book for installation and configuration of WAS V5 on iSeries

Lots of tips to make your work more efficient

IBM WebSphere Application Server for IBM @server™ iSeries (WAS) is an e-business application deployment environment built on open standards-based technology. It is the cornerstone of WebSphere offerings and services. In order to efficiently use WAS on iSeries, customers need to master several skills:

- Installing and configuring the iSeries system for WAS
- Maintaining WAS on iSeries in the most efficient way
- Developing WebSphere applications according to Java 2 Platform, Enterprise Edition (J2EE) specification

This IBM Redbook will help you to gain proficiency in installing, configuring, and administering the WAS environment on iSeries.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-6588-00 ISBN 0738425532