WebSphere. software

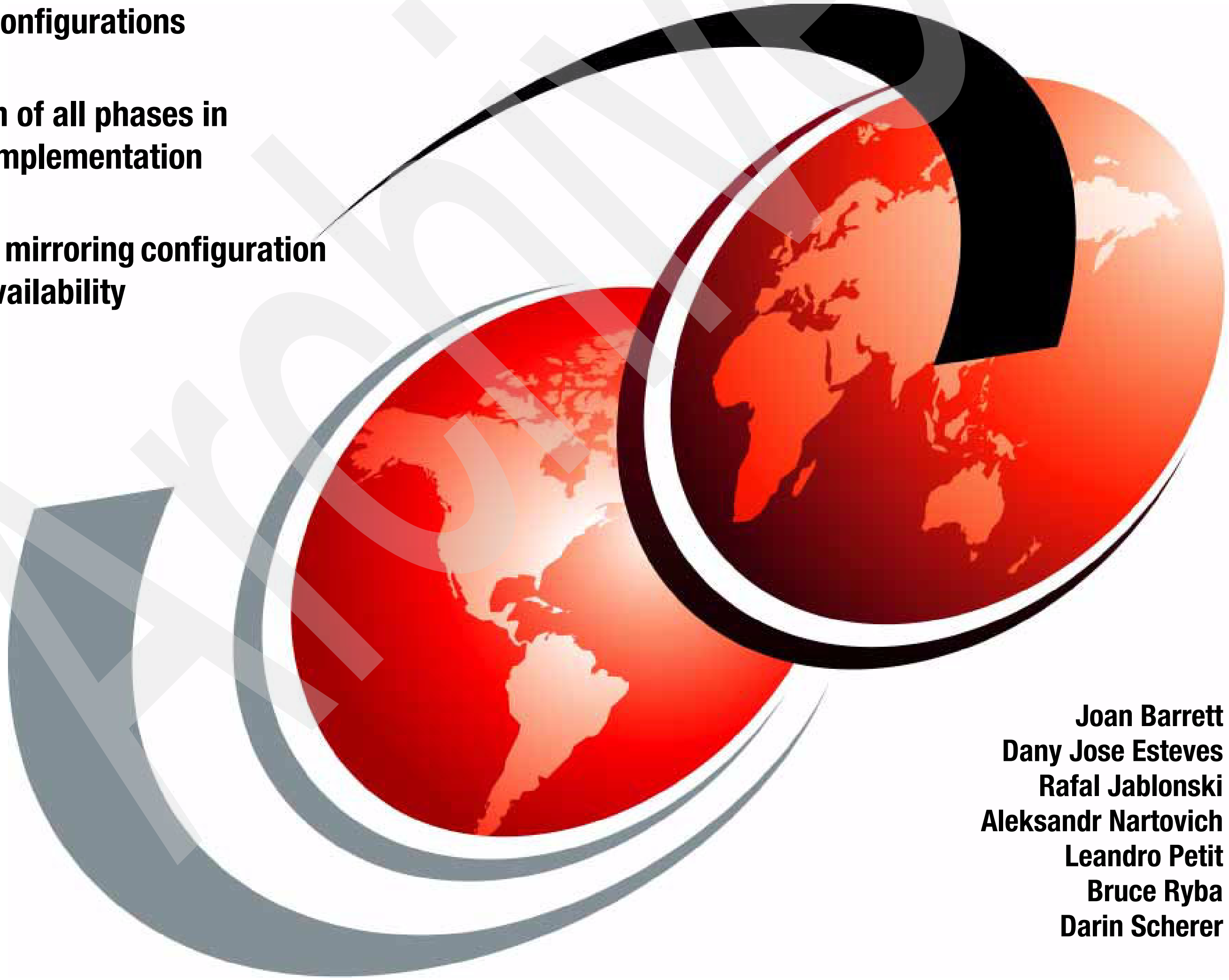
WebSphere Application Server for iSeries V6

Building Advanced Configurations

End-to-end setup instructions for complex configurations

Discussion of all phases in topology implementation

Cross-site mirroring configuration for high availability



Joan Barrett
Dany Jose Esteves
Rafal Jablonski
Aleksandr Nartovich
Leandro Petit
Bruce Ryba
Darin Scherer

Redbooks

ibm.com/redbooks

International Technical Support Organization

WebSphere Application Server for iSeries V6: Building Advanced Configurations

November 2005

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

First Edition (November 2005)

This edition applies to Version 6 of WebSphere Application Server for iSeries (5733-W60).

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®
DB2 Universal Database™
DB2®
eserver®
eserver®
i5/OS™
IBM®
iSeries™
Lotus®
MQSeries®
NetServer™
Operating System/400®
OS/400®
pSeries®
Rational®
Redbooks (logo) ™
Redbooks™
RDN™
Tivoli®
WebSphere®
xSeries®
z/OS®

The following terms are trademarks of other companies:

Enterprise JavaBeans, EJB, Java, JavaBeans, JavaServer, JavaServer Pages, JDBC, JMX, JSP, JVM, J2EE, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

Preface

Many IBM® *@server* iSeries™ clients have passed the initial phase of adoption of IBM WebSphere® Application Server for iSeries. And now with *the* best practices presented in this book, they can build robust, high available solutions, based on WebSphere Application Server.

This IBM Redbook is designed to help system architects, WebSphere administrators, and software developers. It provides a detailed discussion about planning. Plus it provides implementation instructions to help build a complex solution, based on WebSphere Application Server for iSeries. In addition, it provides many useful techniques and tips for such an endeavor.

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 (ITSO), Rochester Center.

Joan Barrett is a software tester at the IBM Toronto Software Lab in Toronto, Canada. For the past five years she has worked testing WebSphere Commerce on both the IBM *@server* iSeries and pSeries® platforms. Prior to software testing, she worked in the TCP/IP and Systems Network Architecture (SNA) communications group, within the iSeries support center, supporting iSeries clients across Canada and in the Caribbean. Her areas of expertise include WebSphere Commerce and WebSphere Application Server.

Dany Jose Esteves serves as a consultant in the IBM Venezuela SLS SW organization and has been in the computer industry for eight years. His area of expertise includes WebSphere and Lotus® family products. He has taught courses on information systems at the Universidad Central de Venezuela for four years and has tutored in the research of information systems.

Rafal Jablonski is a Field Technical Sales Support representative in Poland and the CEMA region and has been with IBM since April 2003. Prior to joining IBM, Rafal had seven years of experience in application development and integration. He started his career as an application developer and has worked with a wide range of development languages and tools. His areas of expertise include OS/400®, Java™ 2 Platform, Enterprise Edition (J2EE™), C and C++, WebSphere Application Server, and MQSeries®. Rafal holds a master degree in electrical engineering from AGH University of Science and Technology in Krakow.

Aleksandr V. Nartovich is a Senior I/T Specialist in the ITSO, Rochester Center. He joined the ITSO in January 2001 after working as a developer in the IBM WebSphere Business Components 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: one in computer science from the University of Missouri-Kansas City and the other in electrical engineering from Minsk Radio Engineering Institute.

Leandro Petit is an Advisory System Software Specialist Professional in IBM Argentina. He has 12 years of experience in iSeries customer support and four years of experience in the WebSphere Application Server and WebSphere Portal Server support areas. He holds a degree in information technology from the Universidad Argentina de la Empresa.

Bruce Ryba is a software engineer at IBM in Rochester, MN. He has 16 years of experience in software engineering. Currently, he is working in the in the Custom Technology Center (CTC) group, specializing in WebSphere customer solutions. His areas of expertise include WebSphere Application Server, WebSphere high availability solutions, and Microsoft® Windows® Integration. Bruce holds a degree in computer science from Mankato State University.

Darin Scherer is a software tester within the iSeries Platform Evaluation Test organization at IBM in Rochester, MN. He has 16 years of experience on the iSeries platform. His areas of expertise include WebSphere Application Server Base and Network Deployment, DB2® Universal Database™ for iSeries, and high-availability technologies. He also partners with IBM consultants to provide guidance for WebSphere high-availability implementations in customer environments.

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

Nadir Amra
Steven D Bussan
Judy I Djugash
Timothy J Klubertanz
Mike McDermott
John McMeeking
Scott Moore
Ron Peterson
Curt Schemmel
Steven Simonson
Frances Stewart
Paul M Swenson
Joseph Zoght
IBM Rochester, Minnesota

Birgit Roehm
IBM Germany

Linda Robinson
ITSO, Raleigh Center

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 email 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

Part 1

Understanding the topologies and their components

This part provides a solid theoretical basis for achieving successful installation and configuration of a complex WebSphere topology. We strongly recommend that you read this part before you jump to the implementation steps presented in Part 2, “Implementation steps” on page 59.

1

Topologies overview

This chapter presents an overview of some possible topologies, their components, and the relationships of those components. After you understand the role of each component and its relationship with the whole topology, you can decide which components to include and which ones to disregard.

Your decision will be influenced by your business requirements and the costs of each component. You must measure costs not only of the money necessary to buy each component but also the skills needed to configure, operate, and maintain the component.

This chapter depicts the following topologies.

- Single machine, single node, one application server topology
- Web server separated topology
- Reverse proxy topology
- Vertical scaling topology
- Horizontal scaling topology
- Horizontal scaling with IP sprayer topology
- Topology with redundancy of several components

1.1 Topology components and relationships

Your Java 2 Platform, Enterprise Edition (J2EE), application will be part of a topology. From a theoretical point of view, a *topology* is a set of components and the relationship between them. We cover those components and relationships in the following sections. Some of the components are optional, where others are required.

1.1.1 Access to the Internet

If your Web application is not an intranet application, you must factor this into consideration. In regard to Internet access, we refer to the bandwidth of the communications link that connects the component that is on the edge of your solution, with the outside world.

1.1.2 Firewall

Every time an organization wants its internal computer network to be connected to the Internet, the organization faces a challenge. It must determine how to keep the internal network safe from the outside intruders, while allowing access to the external world to meet various business requirements.

Theoretically, hackers can break into the company network and steal or damage data, or harm important computer systems or the entire network. To overcome this challenge and protect against hackers, companies build firewalls to give their business legitimate access to the resources outside the company network and to prevent unauthorized external entry into their network.

A firewall is a combination of hardware and software that sits in the entry point to the company network or the point where the company network is connected to the Internet. It monitors the type of traffic that comes into the company network and makes decisions about whether the packet is going to be let in.

A firewall is the TCP/IP equivalent of a security gate at the entrance to your company. All traffic (data packets) must be screened by a security guard (firewall), who then allows only authorized people (packets) to gain entry into the company building (network).

Demilitarized zone

The main purpose of a demilitarized zone (DMZ) configuration is to protect business logic and data from unauthorized access. A typical DMZ configuration includes an outer firewall (protocol firewall) between the public Internet and the Web server or servers that are processing the requests. And it includes an inner firewall (domain firewall) between the Web server and the application servers to which it is forwarding requests. Company data also resides behind the inner firewall.

The area between these two firewalls is called the DMZ. Additional firewalls can further safeguard access to databases that hold administrative and application data.

Protocol firewall (Internet firewall)

A protocol firewall prevents unauthorized access from the Internet to the DMZ. The role of this node is to provide the Internet traffic access only to certain communication protocols, such as HTTP, on certain ports and to block other IP ports.

At least a protocol firewall should be part of your topology. Some security experts say that a protocol firewall is not enough and that a domain firewall should be also included in any configuration. The main argument is that, if the outer firewall is in some way compromised, the inner one can still protect the internal network.

Domain firewall (internal firewall)

The role of the domain firewall allows network traffic that originated only from the DMZ, and not from the Internet, to pass to the private network (intranet). This firewall also provides some filtering from the intranet to the DMZ.

1.1.3 Load balancer

The load balancer, also called *IP sprayer node*, provides horizontal scalability by dispatching HTTP requests among several identically configured Web servers or Web server redirector nodes. The load balancer node can be implemented using the Edge Components provided with the WebSphere Application Server V6 – Network Deployment installation package.

If you plan to include, in your topology, more than one Web server to assure load balancing and failover, a load balancer is a required component.

1.1.4 Reverse proxy

Reverse proxy (or IP forwarding) topology uses a reverse proxy server, such as the one in Edge Components, to receive incoming HTTP requests and to forward them to a Web server. The Web server in turn forwards the requests to the application servers which do the processing. The reverse proxy returns requests to the client, effectively hiding the originating Web server.

A reverse proxy, an optional component, provides one layer more of security to your topology. You can also configure it as a cache to speed up client requests.

1.1.5 Web server

A Web server serves the requests for HTML and other static content. Its function can also be accomplished by the Web container (a component of the application server). However, in most cases, using an external Web server and Web server plug-in as a front-end to a Web container is more appropriate for a production environment.

In a production environment, a Web server is a required component. Use the embedded Web server that is included in WebSphere Application Server only for testing purposes.

1.1.6 Application server

Application servers provide the runtime environment for the application code. They provide containers and services that specialize in enabling the execution of specific Java application components. Each application server runs in its own Java Virtual Machine (JVM™).

An application server is a JVM that runs user applications. The server collaborates with the Web server to return a dynamic, customized response to a client request. Application code, including servlets, JavaServer™ Pages™ (JSP™) files, enterprise beans, and their supporting classes, run in an application server. Conforming to the J2EE component architecture, servlets and JSP files run in a Web container, and enterprise beans run in an Enterprise JavaBeans™ (EJB™) container.

If you need to serve dynamic content (not only static HTML pages), the application server is a required component.

1.1.7 LDAP server

Through a Lightweight Directory Access Protocol (LDAP) server, you can authenticate the users who will access your Web application. LDAP is a directory service, and not a database.

The information in the LDAP directory is descriptive and attribute based. LDAP users generally read the information much more than change it. The LDAP model is based on entries that are referred to as *objects*. Each entry consists of one or more attributes such as a name or address and a type. The types typically consist of mnemonic strings, such as *cn* for common name or *mail* for e-mail address.

Each directory entry also has a special attribute called *objectClass*. This attribute controls which attributes are required and allowed in each entry.

Lightweight Third Party Authentication Protocol

LTPA is an authentication service that supports single signon (SSO) and delegation. Authentication information is carried in LTPA tokens (cookies). It requires an LDAP directory service or custom registry. Operating system user registry is not supported.

1.1.8 Database

A database is a collection of interrelated or independent data items that are stored together to serve one or more applications. From a practical point of view, this means that where the information related to your customers exists, your business and transactions between them reside.

It is key to provide *data resilience*. You must allow the applications to access the data even if the system that originally hosted the data fails.

1.1.9 Network

The machines that are going to be part of your topology will be interconnected through a local area network (LAN) or wide area network (WAN). Do not minimize the impact of link speeds and link utilization rate. A malfunction on the devices that conform the network, such as routers, switches, and hubs, can seriously affect the availability of your services.

If your topology includes logical partitions (LPARs), you can use virtual Ethernet to establish multiple high-speed inter-partition connections. LPAR software allows you to configure up to 16 different virtual LANs. Virtual Ethernet provides the same function as using a 1 GB Ethernet adapter. OS/400 and Linux® partitions can communicate with each other using TCP/IP over the virtual Ethernet communication ports.

1.2 WebSphere Application Server terminology

This section briefly defines some terms that we use to describe the topologies implemented with WebSphere Application Server family of products. We do not define all the terms. For a more in-depth explanation to help you understand these concepts, refer to WebSphere Application Server for OS/400 V6.0 Information Center - Online version

`http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/docs/docws60.html`

1.2.1 Application server node

An application server node (or node) is a grouping of one or more application servers for configuration and operational management. You can have multiple application server nodes on a single machine, but this is not a typical configuration.

1.2.2 Deployment manager

The deployment manager administrates one or more application server nodes in a distributed topology (cell). It communicates (for configuration and administration) with the application server nodes through the node agents.

A deployment manager hosts the administrative console. A deployment manager provides a single, central point of administrative control for all elements of the entire WebSphere Application Server distributed cell. Each cell contains one deployment manager.

Other than the fact that the node agent will not be notified of failed servers on other nodes, the impact of a down (failed or stopped) deployment manager is minimal. If the deployment manager is down, cluster members on the cluster still operate. A performance degradation is possible if it remains down. The consequence of a failed deployment manager is that the node agent is not going to be aware of configuration changes. And neither the console nor wsadmin will be available in the deployment manager node. High availability is improved in WebSphere Application Server V6 through the *High Availability Manager* (HAManager).

1.2.3 Node agent

A node agent identifies the application servers within the application server node in order to be managed by the deployment manager. It is purely an administrative agent and is not involved in application serving functions.

Depending on the configuration, you can have multiple node agents on one single physical machine but only one per application server node. A node agent is created under the covers when you add (federate) a stand-alone node to a cell.

1.2.4 WebSphere profile

Previously known as a *WebSphere instance*, each WebSphere profile has its own configuration files and applications, but shares the binaries with either profile. Each profile is distinguished by its base path, its own directory structure, and its own setupCmdLine script to configure its command-line environment.

1.2.5 Managed node

A managed node is a node that has an application server and a node agent that belongs to a cell.

1.2.6 Web server managed node

A Web server managed node is a Web server that is being managed by the deployment manager. It provides the ability to start and stop the Web server and to automatically push the plug-in configuration file to the Web server. It requires an Application Server Node to be created on the Web server machine. This is commonly used when Web servers are installed behind a firewall where an application server node can be installed.

1.2.7 Web server unmanaged node

A Web server unmanaged node is a Web server configuration that is not being managed by any deployment manager. This is commonly used when Web servers are installed outside a firewall where no application server node can be installed. All Web server implementations prior to Version 6 are of this type.

1.2.8 Cell

A cell is a network of multiple application server nodes. It provides a single logical administration domain and contains only one deployment manager. Deployment managers and nodes can be on the same or different machines or LPARs.

There is flexibility to configure cells according to the administrative requirements. Figure 1-1 shows a possible configuration depicting two cells. One cell has all its components on the same machine or LPAR, and the other cell extends across two different machines or LPARs.

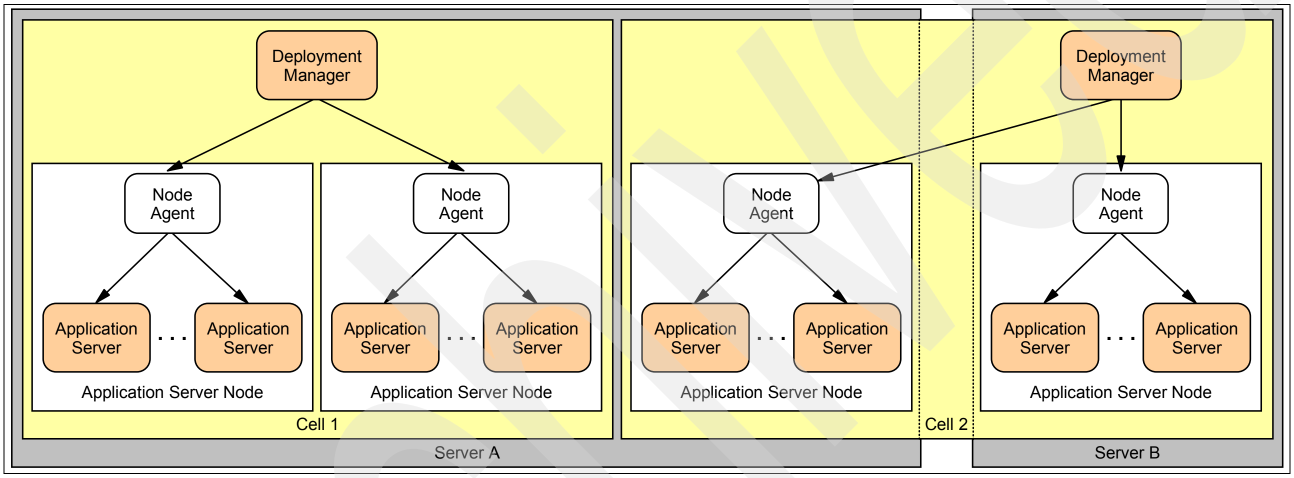


Figure 1-1 Configuration with two cells

1.2.9 Cluster

A cluster is a grouping of application servers called *cluster members*. These cluster members share the same set of J2EE applications. A cluster provides scalability and failover support.

A cell can have zero or more clusters. A cell can span machine or LPAR boundaries (vertical scaling and horizontal scaling) and can span to different operating systems.

Starting in WebSphere Application Server V6.02, you can mix z/OS® nodes with distributed platform nodes and IBM i5/OS™ nodes within the same cell. But you cannot mix z/OS WebSphere cluster members with cluster members running on other platforms when creating a WebSphere cluster. A z/OS WebSphere cluster can only contain cluster members running on z/OS. But, you can have both distributed platform cluster members and i5/OS cluster members to create a heterogeneous WebSphere cluster.

Figure 1-2 shows a cluster that has four cluster members. All cluster members must be part of the same cell. Inside an application server node, some application servers can be members of a cluster and some others are not members.

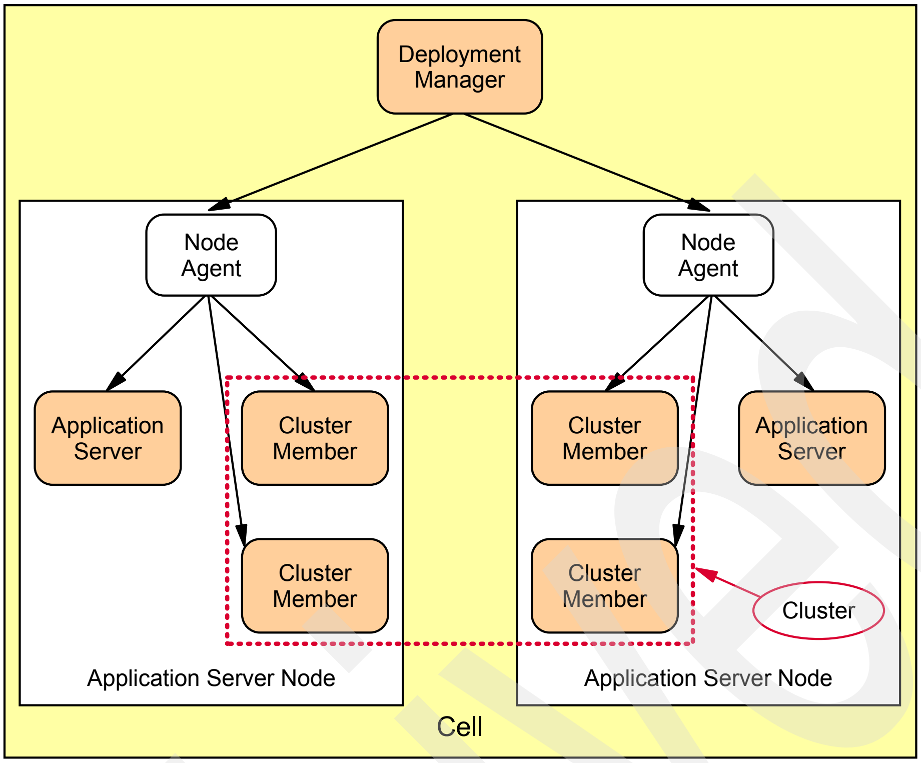


Figure 1-2 Cluster with four cluster members

1.2.10 High Availability Manager

WebSphere Application Server uses a HAManager to eliminate single points of failure. A HAManager is responsible for running key services on available application servers rather than on a dedicated one, such as the deployment manager. The HAManager also provides peer-to-peer failover for critical services by always maintaining a backup for these services. For more information, see "High availability" on page 250.

1.3 From the basic to the most sophisticated topologies

This section provides some configuration examples that help you to fully understand the components, terminology, and the relationship between them in a given topology.

1.3.1 Single machine, single node, one application server topology

The starting scenario for our overview about topologies is a configuration where all the components reside on the same machine or LPAR. The Web server routes requests as appropriate, to the WebSphere Application Server on the same machine for processing (see Figure 1-3).

In this configuration, one application server node contains one Web server, and one application server is running applications. Variations of this topology are possible. Observe that this configuration also includes two firewalls to create a DMZ and a directory server to handle user authentication. A much simpler configuration can be created, without firewalls and a security implementation such as for development or testing purposes.

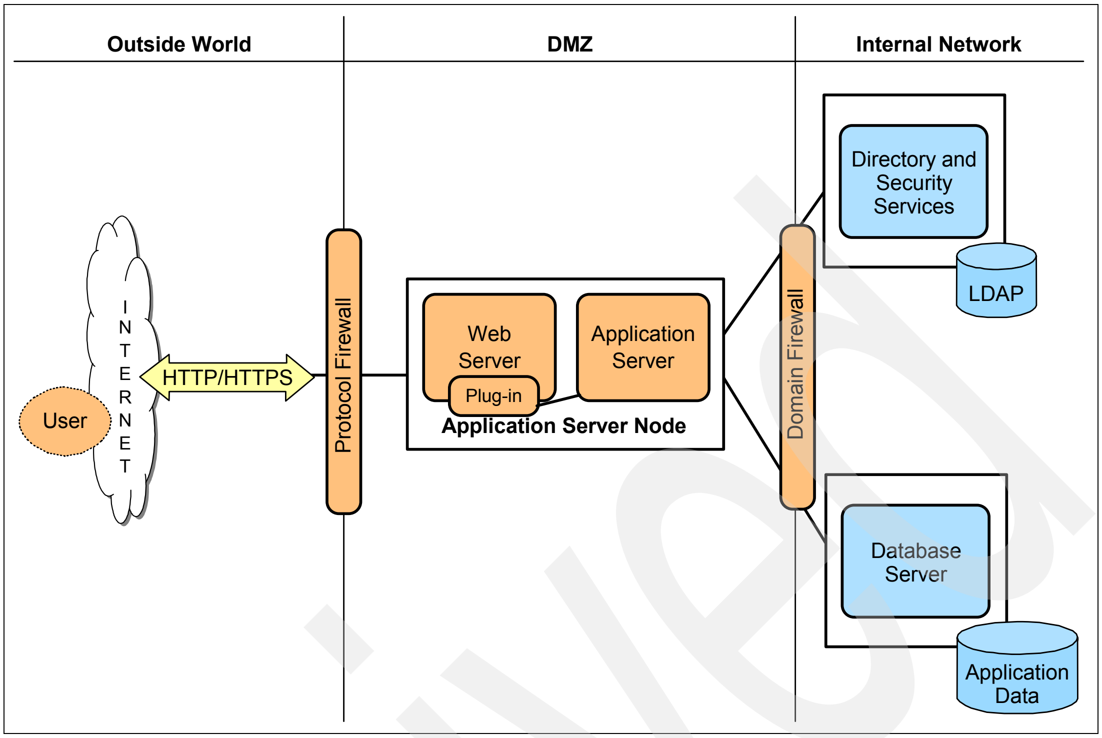


Figure 1-3 One machine, one node, one application server

Advantages

Some good reasons to use a single machine topology are:

- Maintainability (easy to install and maintain)

  This configuration is most suitable as a startup configuration in order to evaluate and test the basic functionality of WebSphere Application Server and related components. The installation is automated by tools supplied with the WebSphere distribution. This configuration is also the easiest to administrate.

- Low cost

  Since the components are located in the same machine, both the cost of the hardware and the cost of maintenance are reduced.

Disadvantages

Consider the following disadvantages of using a single machine topology.

- Performance: Components interdependence and competition for resources

  All components compete for the shared resources (central processing unit (CPU), memory, network, input/output (I/O) and so on). Since components influence each other, bottlenecks or ill-behaved components can be difficult to identify and remediate.

- Security: No isolation between the Web server and application server

  If firewalls and security are not configured, there is no explicit layer of isolation between the components.

- Availability: A single point of failure

  This configuration is a single point of failure. If the hardware, the Web server, or the application server fails, the entire site is not usable.

1.3.2 Web server separated topology

When compared to a configuration where the application server and the Web server are collocated on a single physical server, separation of the application server and the Web server can be used to provide improvement in security, performance, throughput, availability, and maintainability (Figure 1-4).

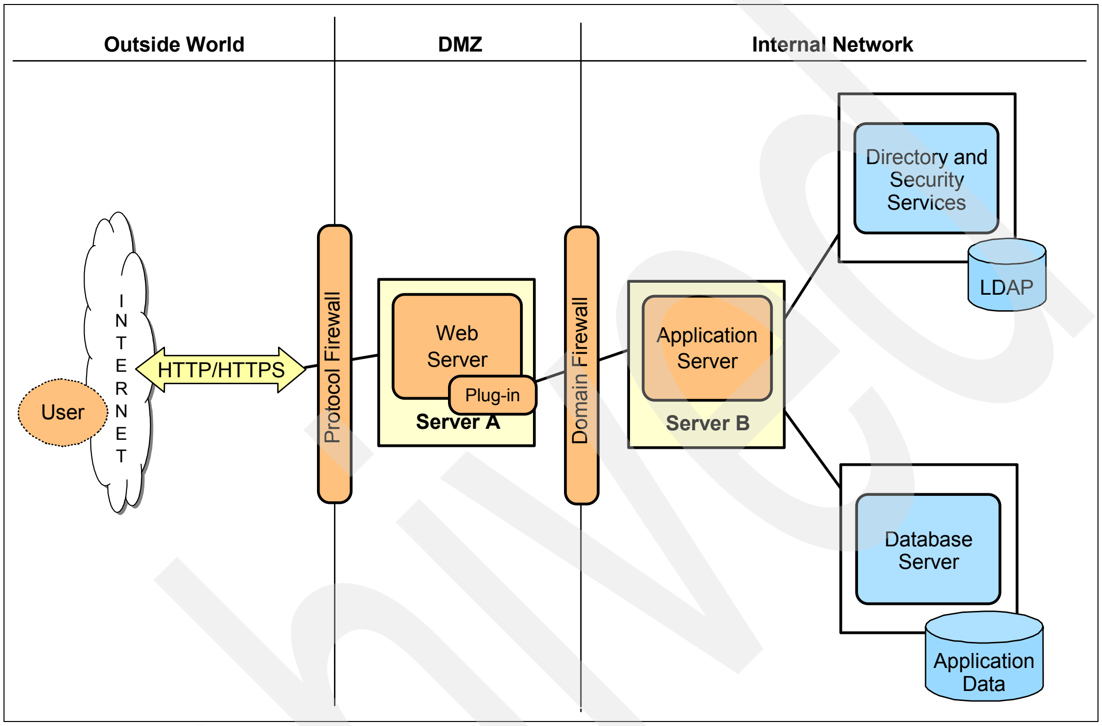


Figure 1-4 Web server separated

The Web server plug-in allows the Web server to route requests to the application server when the servers are physically separated. It uses an Extensible Markup Language (XML) configuration file (plugin-cfg.xml) that contains settings, which describe how to handle and pass on requests to the application server or servers.

Be aware that in this case, the plugin-cfg.xml configuration file is generated on the machine where the application server is installed. Therefore, it must be moved each time it is regenerated from the machine where the application server resides, to the machine where the Web server and the plug-in module are installed. A failure on the Web server can be bypassed by pointing the Domain Name System (DNS) to the machine where WebSphere Application Server is installed. This way, the embedded WebSphere Application Server Web server can replace the Web server (with limited throughput) while the problem is solved.

Advantages

Here are some reasons to separate the Web server from the application server:

- Performance
  - Size and configure servers appropriately to each task.

    By installing components (Web server and application server) on separate machines, each machine can be sized and configured to optimize the performance of each component.

  - Remove resource contention.

    By installing the Web server on a separate machine from the WebSphere Application Server machine, a heavy load of static requests will not affect the resources (CPU, memory, disk) that are available to WebSphere. Nor will such a load affect the ability to service dynamic requests. The same applies when the Web server serves dynamic content using other technologies, such as Common Gateway Interface (CGI).

- Security: Separate Web server and WebSphere interfaces

  To protect the application servers from unauthorized outside access, the separation of the Web server from the application server is often used with firewalls to create a secure DMZ surrounding the Web server. Isolating the Web server on a DMZ protects the application logic and data. It restricts the access from the public Web site to the servers and databases where this valuable information is stored.

  Desirable topologies should neither have databases, nor servers that directly access databases, in the DMZ. WebSphere Application Server stores the configuration data as XML files. Furthermore, an application installed on WebSphere usually needs to access a database. For this reason, we recommend that you do not run WebSphere Application Server in the DMZ.

Disadvantages

Consider the following points when you separate the HTTP server from the application server:

- Maintainability: Configuration complexity

  The configuration file (plugin-cfg.xml) is generated on the WebSphere Application Server machine and must be copied to the Web server machine.

- Performance: Network access possible limitations to performance

  Depending upon the network capacity and remoteness of the Web server, the network response time for communications between WebSphere Application Server and the Web server may limit the application response time. To prevent this, ensure that you have adequate network bandwidth between the Web server and the application server. When collocated on the same server, network response is usually not an issue, but other resource constraints, such as memory and CPU, can limit performance.

- Security considerations: Encrypted transport

  The HTTP plug-in allows encryption of the link between the Web server and the application server, using HTTP over Secure Sockets Layer (SSL) (HTTPS) data encryption. This reduces the risk for attackers to obtain secure information by “sniffing” packets sent between the Web server and application server. A performance penalty usually accompanies such encryption. The HTTP plug-in is suitable for environments that require all network communication to be encrypted, even in a DMZ.

  We recommend that you configure this connection so that the HTTP plug-in and Web container must mutually authenticate each other using public-key infrastructure (PKI). This prevents unauthorized access to the Web container.

1.3.3 Reverse proxy topology

Reverse proxy (or IP forwarding) topology uses a reverse proxy server, such as the one in Edge Components, to receive incoming HTTP requests and to forward them to a Web server. The Web server in turn forwards the requests to the application servers which do the actual processing. The reverse proxy returns requests to the client, effectively hiding the originating Web server (see Figure 1-5).

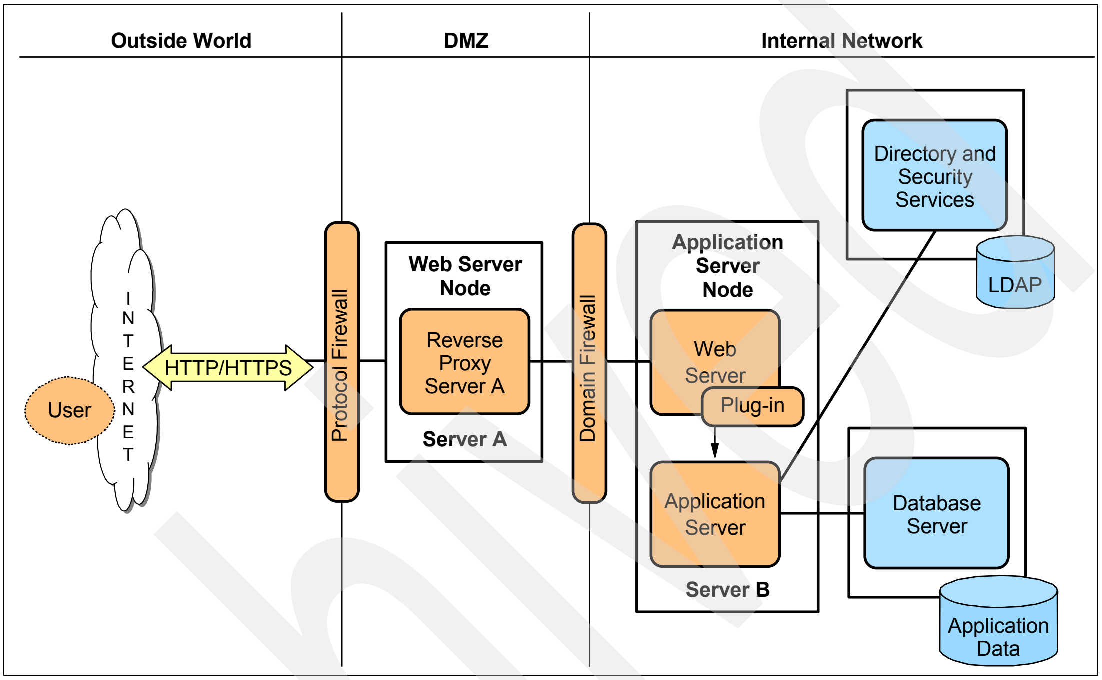


Figure 1-5 Reverse proxy

In this example, a reverse proxy resides in a DMZ between the outer and inner firewalls. It listens on the HTTP port (typically port 80) for HTTP requests. The reverse proxy then forwards those requests to the Web server that resides on the same machine as the application server. After the requests are fulfilled, they are returned through the reverse proxy to the Web client that made those requests. The IP address of the Web server is hidden from the client.

Reverse proxy servers are typically used in DMZ configurations to provide additional security between the public Internet and the Web servers (and application servers) servicing requests. Reverse proxy configurations support high-performance DMZ solutions that require as few open ports in the firewall as possible. For the reverse proxy in the DMZ to access the Web server behind the domain firewall, it requires as little as one port open in the firewall (potentially two ports if using SSL).

Advantages

The advantages of using a reverse proxy server in a DMZ configuration include:

- This is a well-known and tested configuration, so it is easy to implement.
- It is a reliable and fast-performing solution.
- It eliminates protocol switching by using HTTP for all forwarded requests.
- It has no effect on the configuration and maintenance of a WebSphere application.

Disadvantages

The disadvantages of using a reverse proxy server in a DMZ configuration include:

- It requires more hardware and software than similar topologies that do not include a reverse proxy server, making it more complicated to configure and maintain.
- The reverse proxy does not participate in WebSphere workload management.
- It cannot be used in environments where security policies prohibit the same port or protocol being used for inbound and outbound traffic across a firewall.

1.3.4 Vertical scaling topology

Vertical scaling, illustrated in Figure 1-6, refers to configuring multiple application servers on a single machine or LPAR and creating a cluster of associated application servers that all host the same J2EE application or applications.

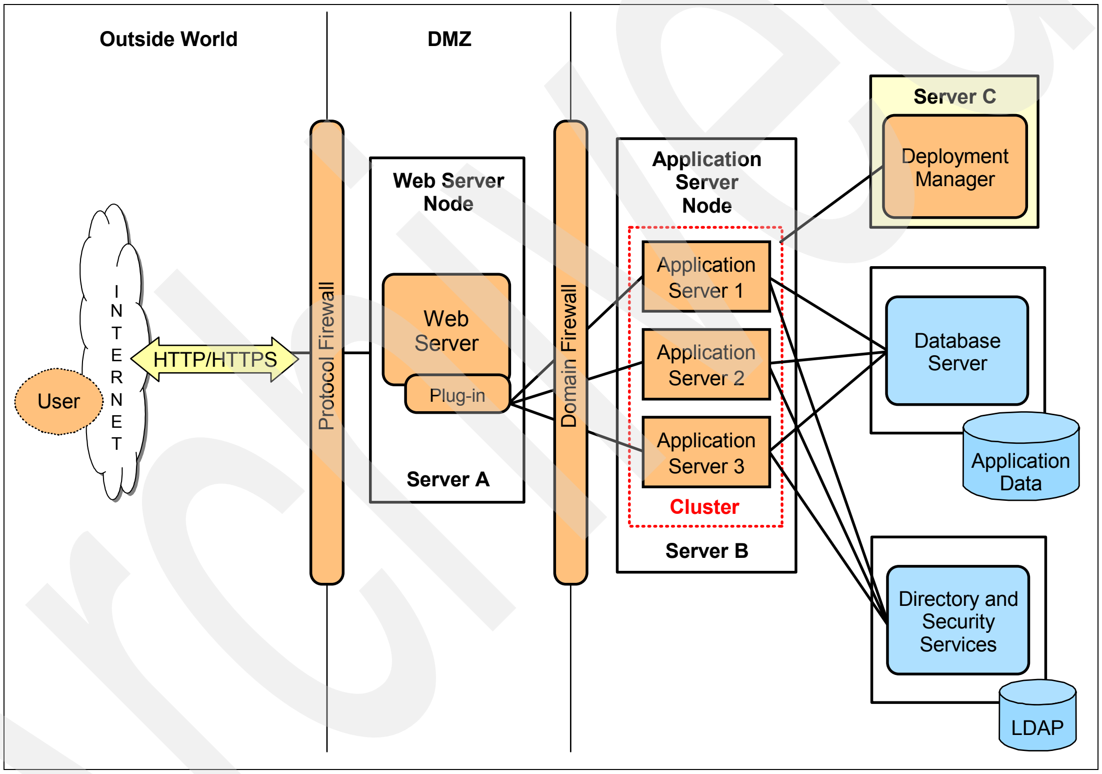


Figure 1-6 Vertical scaling

The vertical scaling example in Figure 1-6 includes a cluster and three cluster members. In this case, the Web server plug-in routes the requests according to the application server's availability. Some basic load balancing is performed at the Web server plug-in level based on a default round-robin algorithm. Vertical scaling can be combined with other topologies to boost performance, throughput, and availability.

Advantages

Vertical scaling has the following advantages:

- Efficient use of machine processing power
- Use of JVM

  Each application server (server1, server2, etc.) runs its own JVM. Each JVM takes a part of the CPU and a part of the memory. Because of JVM architecture and application design, sometimes one JVM cannot fully use all of the CPU or all of the memory. In these cases, more application servers can be created so that each one starts its own JVM, using more CPU and more memory, while improving performance and throughput.

- Load balancing

  Vertical scaling topologies can use WebSphere workload management.

- Process failover

  Vertical scaling can provide failover support among application servers of a cluster. If one application server process goes offline, the other one continues processing client requests.

Disadvantages

Single machine vertical scaling topologies have the drawback of introducing the host machine as a single point of failure in the system.

1.3.5 Horizontal scaling topology

Horizontal scaling exists when the cluster members are located across multiple machines or LPARs. This topology lets a single application span several machines, while presenting itself as a single logical image. Figure 1-7 illustrates the horizontal scaling topology with two application server nodes, each one on separated machines (Server B and Server C). A fourth server, Server D, is where the deployment manager is installed to manage the cluster.

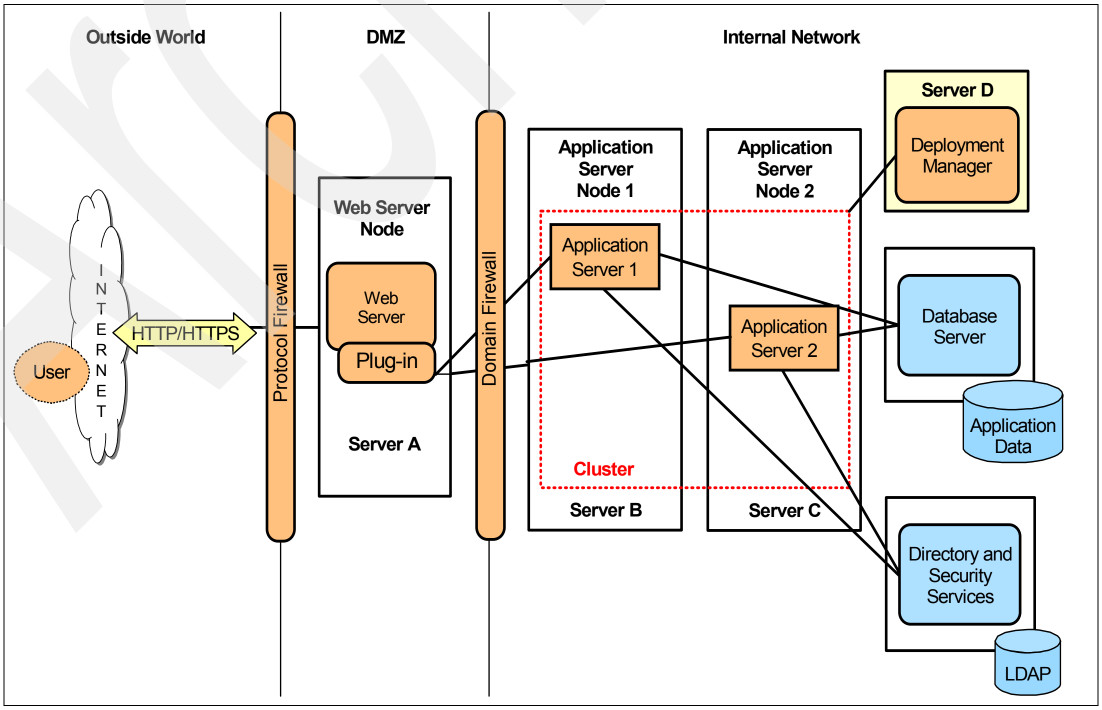


Figure 1-7 Horizontal scaling

The Web server plug-in distributes requests to the cluster members on each server performing some basic load balancing and offering an initial failover. If the Web server (Server A) goes down, then the embedded in WebSphere Application Server Web server of Server B or Server C could be used (limited throughput), while Server A or the Web server on Server A is repaired. Another possibility is to install another Web server on either (or both) Server B or Server C. Then have it configured and off line (stand-by), ready to go to production in the event of any failure on the Web server node.

If any component in the application server node 1 (hardware or software) fails, the application server node 2 can serve requests from the Web server node and vice versa. The Network Dispatcher, part of the Edge Components, can be configured to create a cluster of Web servers and add it to a cluster of application servers (see 1.3.6, “Horizontal scaling with IP sprayer topology” on page 16).

Advantages

Horizontal scaling using clusters has the following advantages:

- Improved throughput

  The use of clusters enables the handling of more client requests simultaneously.

- Improved performance

  Hosting cluster members on multiple machines enables each member to use the machine’s processing resources to avoid bottlenecks and improve response time.

- Hardware failover

  Hosting cluster members on multiple machines isolates hardware failures and provides failover support. Client requests can be redirected to cluster members on other machines if a machine goes offline.

- Application software failover

  Hosting cluster members on multiple nodes isolates application software failures and provides failover support if an application server stops running. Client requests can be redirected to cluster members on other nodes.

Disadvantages

Horizontal scaling using clusters has the following disadvantages:

- Higher hardware and software costs
- More complex maintenance
- Maintenance of application servers on multiple machines

1.3.6 Horizontal scaling with IP sprayer topology

Load balancing products can be used to distribute HTTP requests among Web servers that are running on multiple physical machines or LPARs. The Network Dispatcher, part of WebSphere Edge Components, is an IP sprayer that performs intelligent load balancing among Web servers. It is based on server availability and workload as the main selection criteria to distribute the requests.

Figure 1-8 illustrates a horizontal scaling configuration that uses an IP sprayer on the load balancer node to redistribute requests between Web servers on multiple machines.

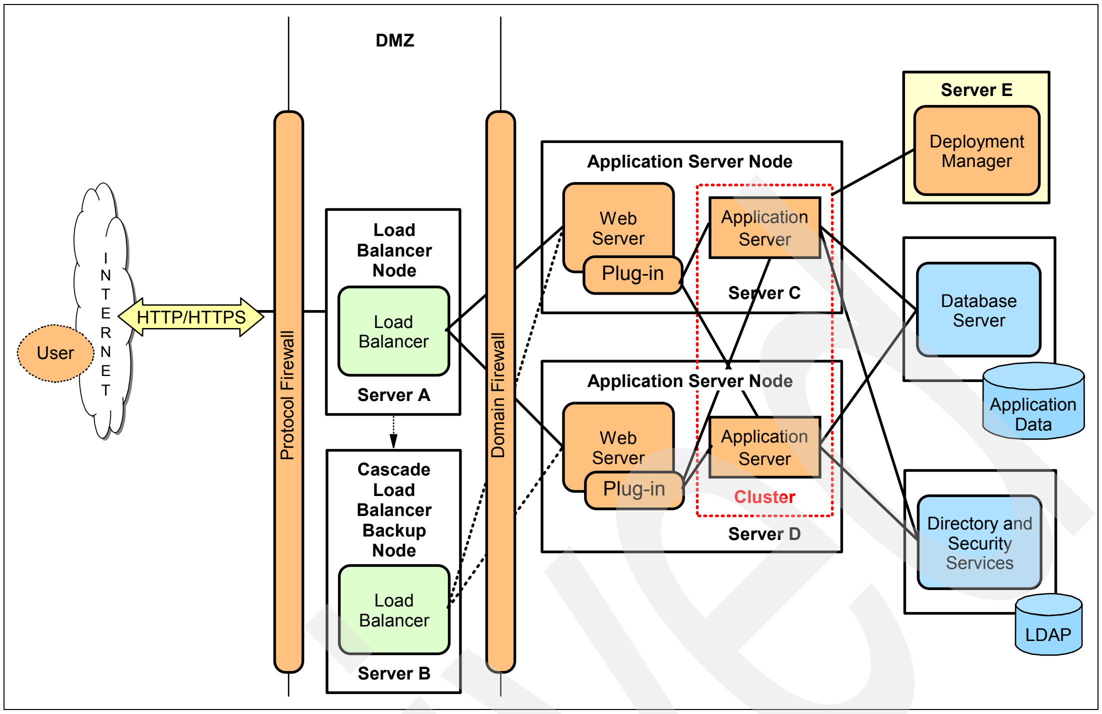


Figure 1-8 IP sprayer and horizontal scaling

The load balancer node sprays Web client requests to the Web servers. The load balancer is configured in cascade. The primary load balancer communicates to its backup through a heart beat to perform failover if needed. It also eliminates the load balancer node as a single point of failure. Both Web servers perform load balancing and failover between the application servers (cluster members) through the Web server plug-in module and configuration file. If any component on Server C or Server D fails, the others can continue receiving requests.

Advantages

Using an IP sprayer to distribute HTTP requests has the following advantages:

- Improved server performance by distributing incoming TCP/IP requests (in this case, HTTP requests) among a group of Web servers
- The use of multiple Web servers, which increases the number of connected users that can be served at the same time
- Elimination of the Web server as a single point of failure

  Used in combination with WebSphere workload management, this topology eliminates the application servers as a single point of failure.
- Improved throughput and performance by maximizing CPU and memory utilization

Disadvantages

Using an IP sprayer to distribute HTTP requests has the following disadvantages:

- Extra hardware and software are required for the IP sprayer servers.
- As with the previous topology, extra maintenance of multiple machine is required.

1.3.7 Topology with redundancy of several components

Redundancy of components eliminates or minimizes the existence of a single point of failure. Figure 1-9 shows a topology where most components have some kind of redundancy, such as a load balancer backup node in cascade with the primary load balancer node, clustered Web servers, and application servers.

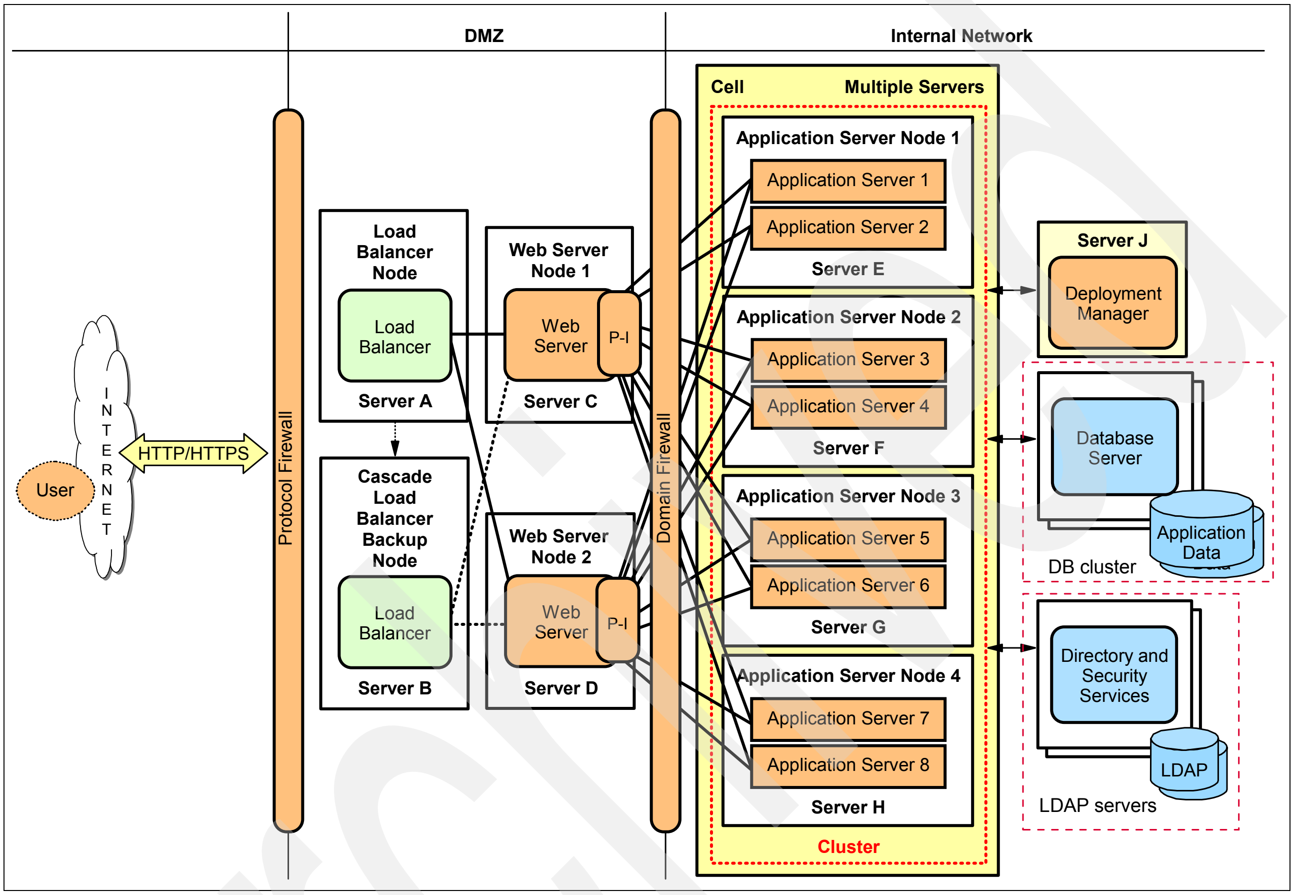


Figure 1-9 Topology with redundancy of several components

The redundant components in this example include:

- Two load balancers

  The one on Server A is the primary load balancer. It is synchronized, through a heart beat, with a backup load balancer in cascade that is in stand-by mode on another machine, Server B.

- Two Web servers

  Both Web servers receive requests from the load balancer and share the requests that come from the Internet. Each one is installed on a different machine.

- A cluster of application servers

  The cluster implements vertical and horizontal scaling.

- Eight cluster members, two on each application server node
- Two database servers

  The database servers use a high availability software product. One copy of the database is used, and the other one is a replica that will replace the first one if it fails.

- Two LDAP servers

  The LDAP servers use a high availability software product. One copy of the directory is used, and the other one is a replica that will replace the first one if it fails.

Advantages

This topology is designed to maximize performance, throughput, and availability. It incorporates the benefits of the other topologies that have already been discussed in this chapter.

- A single point of failure is eliminated from the load balancer node, Web server, application server, database server, and LDAP server, due to redundancy of those components.
- The topology provides both hardware and software failure isolation. Hardware and software upgrades can be easily handled during off-peak hours.
- Horizontal scaling is done by using both the IP sprayer (for the Web server nodes) and the application server cluster to maximize availability.
- Application performance is improved by using several techniques.
  - Hosting application servers on multiple physical machines to boost the available processing power
  - Using clusters to vertically scale application servers, which makes more efficient use of the resources of each machine
- Applications with this topology can use workload management techniques. In this example, workload management is performed through:
  - WebSphere Edge Component - Network Dispatcher to distribute client HTTP requests to each Web server
  - WebSphere Application Server - Network Deployment workload management feature to distribute work among clustered application servers

Disadvantages

This combined topology has one disadvantage in that you must evaluate the costs in hardware, configuration, and administration in relationship to performance, throughput, and reliability.

2

Planning for a new topology

This chapter describes the stages that characterize the process of planning for a new topology. In these stages, you consider such aspects as business needs, the relationship between the business and technology goals, characteristics of the new solution, and the relationship with the new topology.

This chapter does not attempt to present a methodology to plan for a new topology. There are multiple methodologies for this and volumes dedicated to best practices.

This chapter covers the following topics:

- Relating business requirements and technology requirements
- Understanding the requirements of the solution
- Selecting the topology
- Reviewing of the products mapping

2.1 Overview of topology selection process

Planning for a new topology is a consequence of market needs that are translated into business needs. These necessities of business will be satisfied by IT means.

In general a new planning project starts with a group of ideas. Little is known about specific details, especially as they relate to the topology. Figure 2-1 shows the four main aspects that you must consider at the time of planning a new topology. These aspects are:

- Key business
- Information view
- Functional view
- Operational view

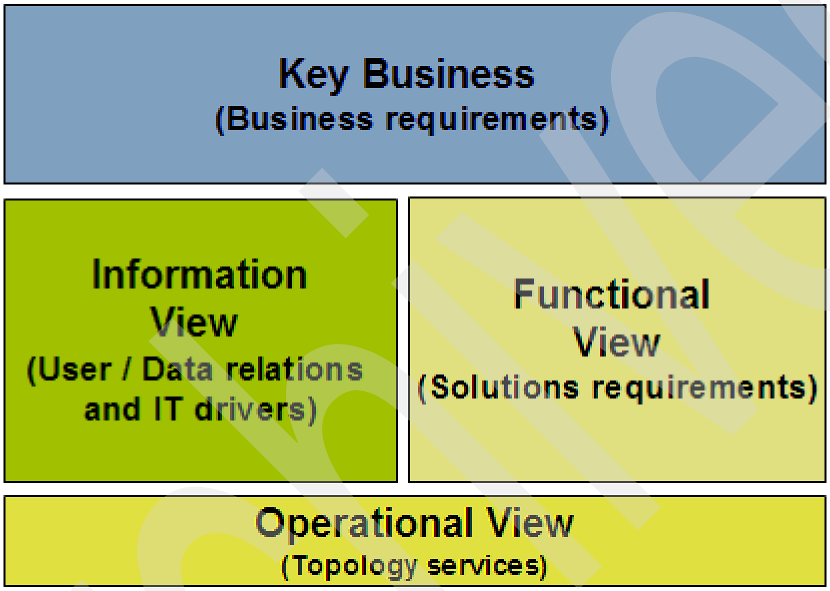


Figure 2-1 Main aspects of planning a new topology

The four stages of the selection process, of a new topology, are based on the Patterns for e-business that IBM developed, based on our extensive experience with our clients. To learn about IBM Patterns for e-business, see the *IBM Patterns for e-business* Web site at:

http://www.ibm.com/developerworks/patterns/index.html

The planning stages are:

1. Relating business requirements and technology requirements

   First, you must describe the key business purposes of the solution. Consider the relationship between the business, the users, and the data. After you identify the key business purpose of a solution, you need to understand the relationship between the key business and IT drivers.

2. Understanding the requirements of the solutions

   Next, you must answer the question: How do applications support the required functionality?

3. Selecting the topology

   Then you must select the topology and its components. In this stage, you relate the requirements of the solution with the services offered by the new topology.

4. Reviewing the products mapping

   After you select the topology, define its logical components, and identify the services that they offer, relate these components to the specific products that will enable you to implement the solution.

2.2 Patterns for e-business

IBM has created a variety of documentation to help customers to understand the process of building a solution, from identifying the business requirements and to selecting a topology and products that map to it. IBM has found that, in most cases, there are many similarities in a group of solutions. Based on this analysis, IBM identified these similarities as *patterns*.

Patterns for e-business are a group of reusable assets that can help speed the process of developing Web-based applications. The Patterns leverage the experience of IBM architects to create solutions quickly, whether for a small local business or a large multinational enterprise.

2.2.1 e-business solution

An *e-business solution* is a set of technologies that allows the company to improve its processes and communication with employees, clients, suppliers, and business partners. Each solution possesses certain characteristics. Table 2-1 details some of the characteristics of an e-business solution.

Table 2-1 Characteristics of an e-business solution

Characteristic	Description
Reach	Across departments, enterprise, geographic, and national borders
Architecture topology	Deployed using thin a client architecture with clients connected to backend tiers
Client hardware	Deployed on workstations that could be PCs, network computers, personal digital assistants (PDAs), voice response units (VRUs), and set-up devices
Server hardware	One or more Web servers with additional components such as application servers added as needed
Network	Local area network (LAN), wide area network (WAN), virtual private networks (VPNs), Internet, leased lines and dial-up connections
Standards	Open technologies
Programming model	Processing done mostly by the server; clients use an event

e-business solution types

The e-business solutions can be grouped into four categories.

- Self service
- Collaboration
- Information aggregation
- Extended enterprise solutions

Self service

Also known as the *user-to-business solution*, self service addresses internal and external users interacting with enterprise transactions and data. Table 2-2 lists examples of self-service solutions grouped by industry.

Table 2-2 Examples of self-service solutions

Type of organization	Type of solution
Insurance	Locate a nearby office Locate brokers or agents Financial planner and insurance needs analysis tool Portfolio summary Policy summary and details Claims submission and tracking Online billing
Manufacturing	Review required parts or services Locate service centers Register for training classes Submit or track orders
Banking	View account balances View recent transactions Pay bills or transfer funds Stop payments Manage bank card
Telecommunication	Review account statements Paying bills online Change personal profile Add, change, or remove services (for example, call waiting or caller ID) Submitting service requests
Government	Submit tax returns Renew automobile licenses Download forms or applications Submit forms or applications

Collaboration

Collaboration addresses the interactions and collaborations between users. This type of e-business solution is intended for small or extended teams that need to work together to achieve a common goal. Table 2-3 lists examples of collaboration solutions.

Table 2-3 Examples of collaboration solutions

Type of organization	Type of solution
Large enterprises, small and medium businesses	E-mail Bulletin boards News groups Instant messaging Team rooms Online meetings Ad hoc workflow

Information aggregation

The information aggregation solution, also known as the *user-to-data solution*, allows users to access and manipulate data that is aggregated from multiple sources. This solution captures the process of taking large volumes of data, text, images, video and so on and using tools to extract useful information from them. Table 2-4 lists examples of information aggregation solutions.

Table 2-4 Information aggregation example

Type of organization	Type of solution
Large enterprises, small and medium businesses	Strategy and planning Senior management decisions Business strategy planning Transaction summary and tracking Statistical analysis
	Marketing analysis Identifying markets Identifying prospective customers Contacting customers Managing the marketing process
	Sales management Following through on marketing leads Managing the sales process Identifying cross-selling opportunities
	Service management Customer service analysis Supplier management Business support services
	Product analysis Product competitiveness analysis Brand management

Extended enterprise

Extended enterprise, also known as the *business-to-business (B2B) solution*, addresses the interactions and collaborations between business processes in separate enterprises. This type of solution applies to organizations that need to implement programmatic interfaces to interconnect the enterprises' applications. Table 2-5 lists examples of such solutions.

Table 2-5 Examples of extended enterprise solutions

Type of organization	Type of solution
Travel	Checking flight or room availability Making or modifying reservations
Retail	Checking supplier inventory Placing replenishment orders Paying suppliers automatically
Financial	Transferring payments Checking account balances Obtaining credit information Loan origination Processing securities

Type of organization	Type of solution
Telecommunication	OSS integration Cross organization order management Managed service provider interconnect
Manufacturing	Supply chain planning Supply chain execution Vendor managed inventory

2.2.2 Application patterns

After you identify the type of e-business solution that suits your requirements, you must develop a high-level view of the architecture of the solution. This is the process of mapping your business goals to the technology requirements.

To simplify this step, IBM has created multiple application patterns for each type of e-business solution. We give examples of these application patterns for the *self-service solution*.

The self-service solutions include these application patterns:

- Stand-alone Single Channel
- Directly Integrated Single Channel
- As-is Host
- Customized Presentation to Host
- Router
- Decomposition
- Agent

In the following sections, you see how each pattern applies to your business requirements. Review the decision-making tables to help you select an appropriate application pattern.

Stand-alone Single Channel

The Stand-alone Single Channel application pattern provides a structure for applications that have no current need for integration with other systems, but need only to focus on one delivery channel. Figure 2-2 illustrates the flow of this application pattern.

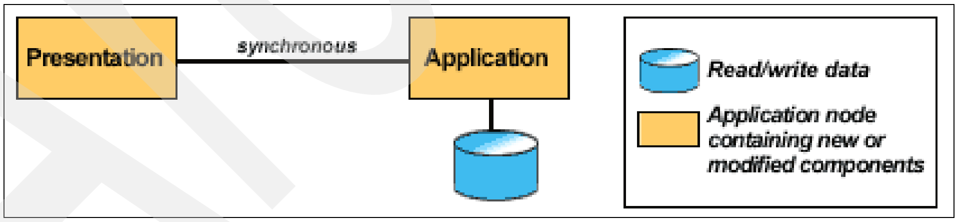


Figure 2-2 Stand-alone Single Channel application pattern

Business and IT drivers that map into this pattern are:

- Time to market
- Minimize application complexity

This is the simplest of all application patterns that automate the self-service business pattern. In this application pattern, the application is divided into two logical tiers, and presentation logic is separated from business logic. This separation ensures the maintainability of the developed application and support of the thin clients.

- The *presentation tier* is responsible for all the user interface-related logic, including data formatting and screen navigation.
- The *application tier* is responsible for implementing business logic and for accessing data from a local database.

Directly Integrated Single Channel

The Directly Integrated Single Channel application pattern provides a structure for applications that need one or more point-to-point connections with back-end applications, but only need to focus on one delivery channel. Figure 2-3 illustrates this application pattern.

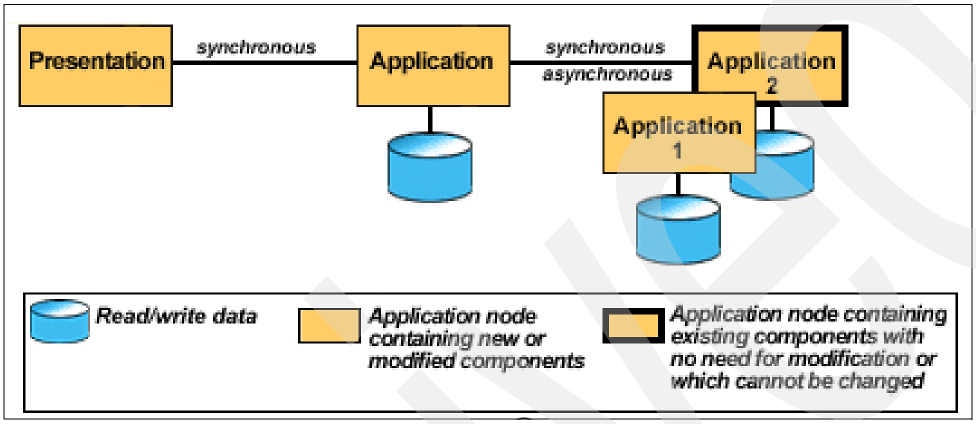

Presentation
synchronous
Application
synchronous
asynchronous
Application 2
Application 1
Read/write data
Application node containing new or modified components
Application node containing existing components with no need for modification or which cannot be changed

Figure 2-3 Directly Integrated Single Channel application pattern

Business and IT drivers that map into this pattern are:

- Improvement of organizational efficiency
- Reduction of latency of business events
- Leverage of existing skills
- Leverage of the existing investment
- Back-end application integration

The Directly Integrated Single Channel application pattern extends the Stand-alone Single Channel application pattern by using point-to-point connections to back-end applications and databases. In this application pattern, applications are divided into at least three different logical tiers: presentation, Web application, and back-end application. These tiers make a logical three-tier architecture. Data can reside on the second and third tiers.

- The *presentation tier* is responsible for all the presentation logic of the application.
- The *Web application tier* is responsible for implementing some of the business logic and for accessing back-end application logic and data. Usually new data, such as user profiling information, resides on this tier.
- The *back-end application tier* can represent a new application, a modified existing application, or an unmodified existing application. The data of existing systems resides on this tier and is most likely accessible only through the existing back-end application. It is important to note that multiple back-end applications can be accessed by the same Web application tier.

As-is Host

The As-Is Host application pattern provides wider intranet access to existing host applications. These applications may previously have only been available to employees through green-screen devices or PCs with emulators. Figure 2-4 illustrates the flow of this application pattern.

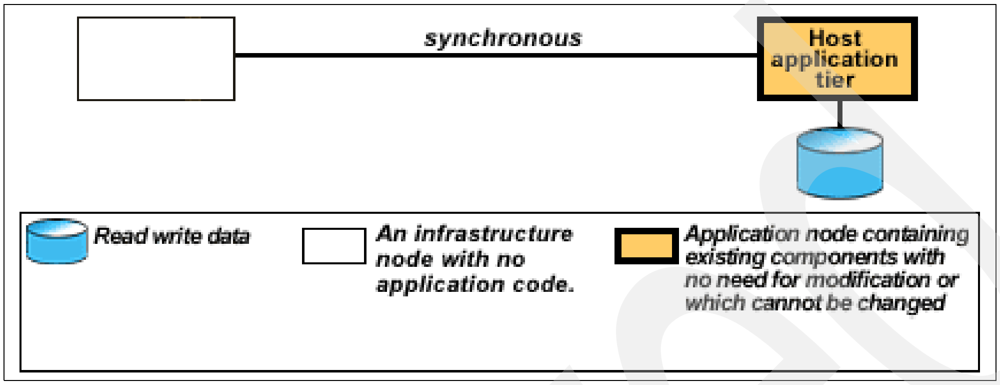

synchronous
Host application tier
Read write data
An infrastructure node with no application code.
Application node containing existing components with no need for modification or which cannot be changed

Figure 2-4 As-Is Host application pattern

Business and IT drivers that map into this pattern are:

- Time to market
- Improvement of organizational efficiency
- Minimization of total cost of ownership (TCO)
- Leverage of existing skills
- Leverage of the existing investment
- Back-end application integration

As shown in Figure 2-4, the As-Is Host application pattern has two logical tiers.

- The *empty presentation tier* represents an off-the-shelf middleware component such as IBM WebSphere Host Access Transformation Services (HATS) that can be used to provide browser-based access to host applications. This tier does not include any custom built code.
- The *host application tier* represents the existing back-end application.

Customized Presentation to Host

The Customized Presentation to Host application pattern helps to provide a more user-friendly interface to existing host applications without changing the underlying application. Figure 2-5 illustrates the flow of this application pattern.

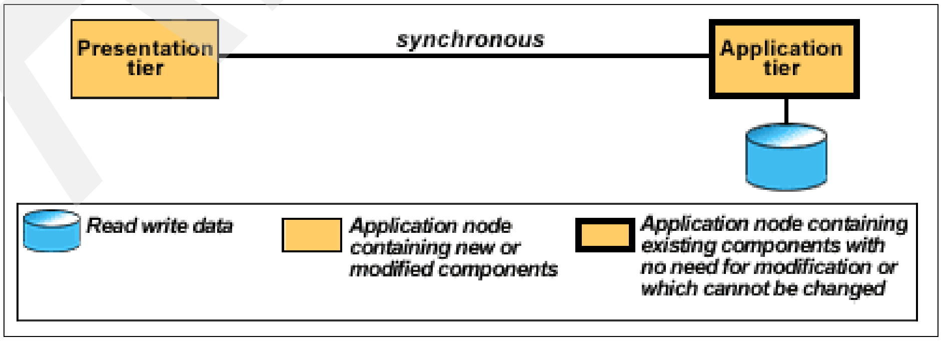

Presentation tier
synchronous
Application tier
Read write data
Application node containing new or modified components
Application node containing existing components with no need for modification or which cannot be changed

Figure 2-5 Customized Presentation to Host application pattern

Business and IT drivers that map into this pattern are:

- Time to market
- Improvement of organizational efficiency
- Minimization of application complexity
- Minimization of TCO
- Leverage of existing skills
- Leverage of existing investment
- Back-end application integration

As shown in Figure 2-5, the Customized Presentation to Host application pattern implements a thin client that provides a customized presentation to an existing host application, while keeping the back-end system as is. It has two logical tiers.

- The *customized presentation tier* accesses the host application and presents the results in a rich graphical user interface (GUI) format.
- The *host application tier* represents the existing back-end application.

Router

The Router application pattern provides a structure for applications that require the intelligent routing of requests from multiple, delivery channels to one of multiple, back-end applications. Figure 2-6 illustrates the flow of the Router application pattern.

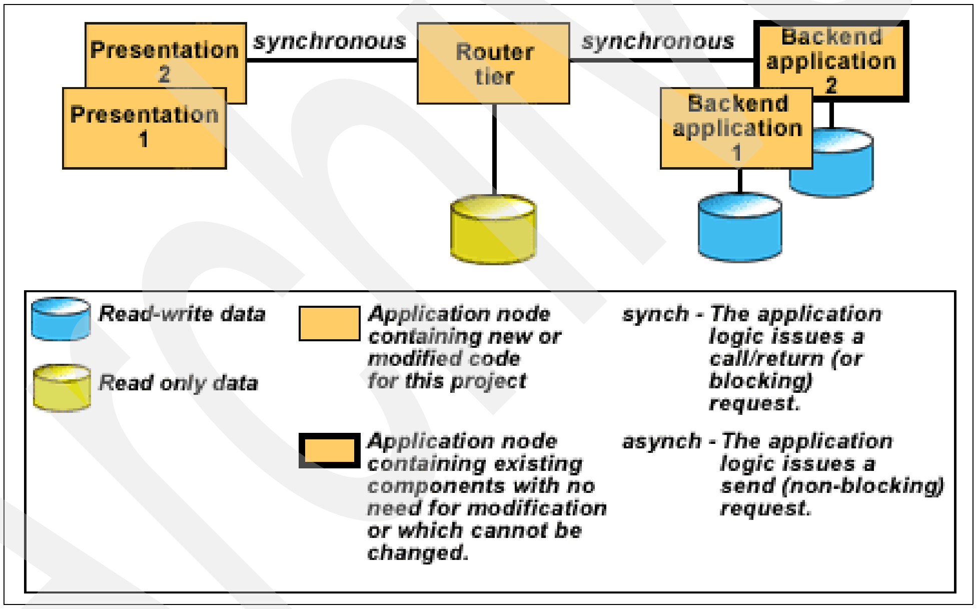

Presentation 2
Presentation 1
synchronous
Router tier
synchronous
Backend application 2
Backend application 1
Read-write data
Read only data
Application node containing new or modified code for this project
Application node containing existing components with no need for modification or which cannot be changed.
synch - The application logic issues a call/return (or blocking) request.
asynch - The application logic issues a send (non-blocking) request.

Figure 2-6 Router application pattern

Business and IT drivers that map into this pattern are:

- Reduction of latency of business events
- Easy to adapt during mergers and acquisitions
- Integration across multiple delivery channels
- Minimization of TCO
- Leverage of existing skills
- Leverage of existing investment
- Back-end application integration
- Minimization of enterprise complexity
- Maintainability
- Scalability

As shown in Figure 2-6, the Router application pattern is divided into at least three logical tiers.

- Unlike the previous self-service application patterns, the *presentation tiers* of this application pattern can support many different presentation styles, including the Internet, call centers, kiosks, and VRUs.
- The *router tier* receives requests from multiple presentation components and intelligently routes them to the appropriate back-end transactions. In doing so, this tier may use a read-only database to look up routing rules. In addition, the router may be responsible for message transformation, protocol conversion, the management of different levels of security, and session concentration. In most cases, the router tier implements minimal business logic. This routing capability can be used to rout requests from one back-end system to the other.
- The majority of the business logic for this application pattern is concentrated in the *back-end application tier*.

Decomposition

The Decomposition application pattern extends the hub and spoke architecture provided by the Router application pattern. It decomposes a single, compound request from a client into several, simpler requests and intelligently routes them to multiple back-end application. Typically the responses from these multiple back-end applications are recomposed into a single response and sent back to the client. Figure 2-7 illustrates this application pattern.

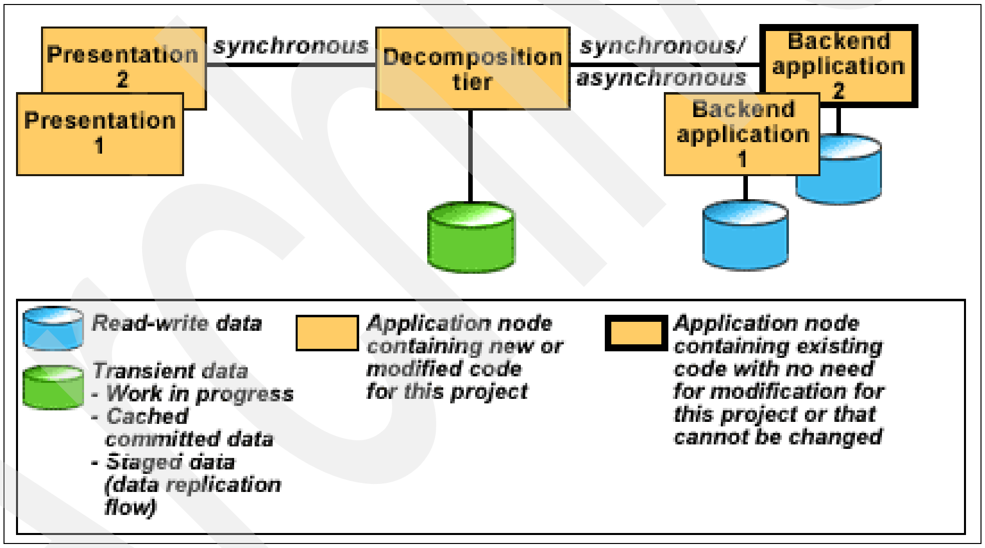

Presentation 2
Presentation 1
synchronous
Decomposition tier
synchronous/ asynchronous
Backend application 2
Backend application 1
Read-write data
Transient data
- Work in progress
- Cached committed data
- Staged data (data replication flow)
Application node containing new or modified code for this project
Application node containing existing code with no need for modification for this project or that cannot be changed

Figure 2-7 Decomposition application pattern

Business and IT drivers that map into this pattern are:

- Improvement of organizational efficiency
- Reduction of the latency of business events
- Easy to adapt during mergers and acquisitions
- Integration across multiple delivery channels
- Unified customer view across Lines of Businesses (LOB)
- Minimization of TCO
- Leverage of existing skills
- Leverage of existing investment
- Back-end application integration

- Minimization of enterprise complexity
- Maintainability
- Scalability

As shown in Figure 2-7, this application pattern is divided into three logical tiers.

- The *presentation tier* is responsible for all the user interface-related logic, including data formatting and screen navigation. It can support many different presentation styles, including the Internet, call centers, kiosks, and VRUs.
- The *decomposition tier* supports most of the services provided by the router tier in the Router application pattern, including intelligent routing of requests, protocol conversion, security, and session concentration.

  In addition, it implements the intelligence to break down a single request received from a presentation client into several, simpler requests which it routes to multiple back-end applications. In doing so, it typically uses a local Work In Progress (WIP) database to store routing, decomposition, and recomposition rules, and to cache the results from multiple back-end applications until a recomposition of the desired response has been generated.

  The decomposition tier implements significantly more business logic than a router tier. Such business logic focuses on providing a unified customer-centric view.
- The majority of the product and function-specific business logic is still concentrated in the *back-end application tier*. Some of these back-end applications are highly available, and scalable online transaction processing systems and others are batch applications.

Agent

The Agent application pattern structures an application design that provides a unified customer-centric view that can be exploited for mass customization of services and for cross-selling purposes. Figure 2-8 illustrates the flow of this application pattern.

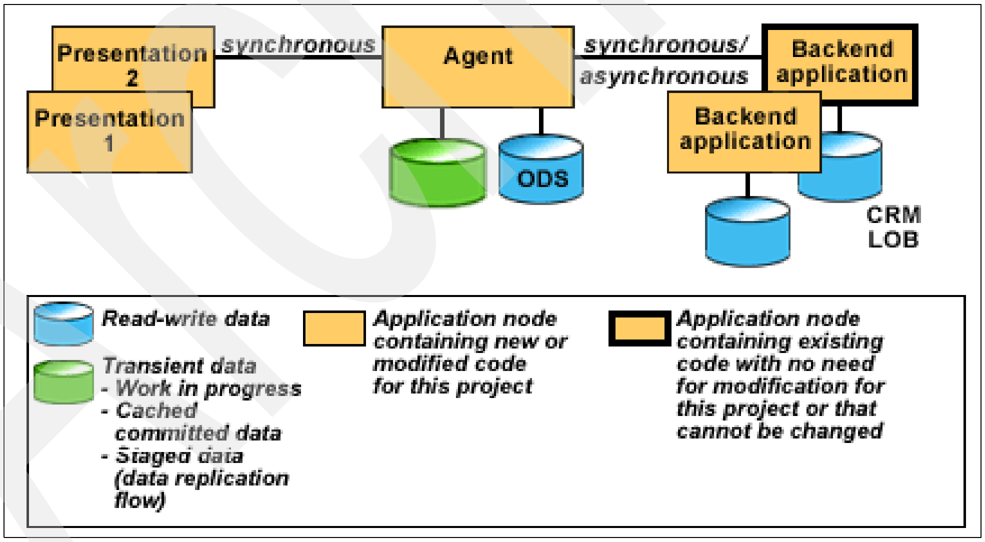

Presentation 2
Presentation 1
synchronous
Agent
ODS
synchronous/ asynchronous
Backend application
Backend application
CRM
LOB
Read-write data
Transient data
- Work in progress
- Cached committed data
- Staged data (data replication flow)
Application node containing new or modified code for this project
Application node containing existing code with no need for modification for this project or that cannot be changed

Figure 2-8 Agent application pattern

Business and IT drivers that map into this pattern are:

- Improvement of organizational efficiency
- Reduction of the latency of business events
- Easy to adapt during mergers and acquisitions
- Integration across multiple delivery channels
- Unified customer view across LOBs

- Support of effective cross selling
- Mass customization
- Minimization of TCO
- Leverage of existing skills
- Leverage of existing investment
- Back-end application integration
- Minimization of enterprise complexity
- Maintainability
- Scalability

As shown in Figure 2-8, this application pattern is divided into three logical tiers.

- The *presentation tier* of this application pattern can support many different presentation styles, including the Internet, call centers, kiosks, and VRUs.
- The *agent tier* supports all the services provided by the decomposition tier in the Agent application pattern. It also dynamically builds a consolidated view of the user's relationship with the organization. It uses this to identify ways to perform mass customization of the organization's goods and services to fit this individual user. This results in pushing an additional browser instance in front of the user so the user can accept or reject the customized offer, before continuing with their original task.
- The majority of the product and function specific business logic is still concentrated in the *back-end application tier*.

2.3 Mapping business requirements to the application patterns

Mapping the business requirements to the application patterns is the first stage in the process of planning a new topology. Figure 2-9 highlights two main aspects that the clients must consider at this stage. These aspects are the key business and information view.

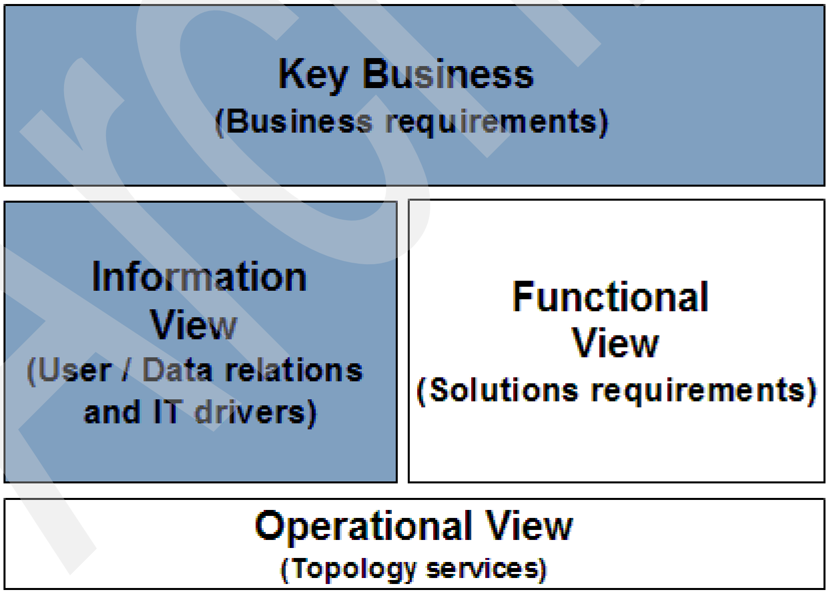


Figure 2-9 Planning for a new topology process: Main aspects for the first stage

2.3.1 How to approach this step

To map your business requirements to the application patterns, you must perform these tasks:

1. Identify the business environment.
2. Identify the business drivers.
3. Identify the business processes.
4. Identify the information view.

Identifying the business environment

The business environment is usually represented in terms of the various business entities. These are the entities that the enterprise or business has to interact with when accomplishing its business functions. Some of the business entities could be:

- Consumers
- Distributors
- Suppliers
- Bankers
- Business partners
- Influencers
- Clearinghouses
- Regulatory bodies

This step helps to:

- Understand the external relationships with businesses and individuals in the marketplace
- Determine the set of business processes

Identifying the business drivers

Some of the business drivers could be:

- Decrease the time to market
- Improve the organizational efficiency
- Reduce the latency of business events
- Integrate across multiple delivery channels
- Provide a unified customer view across lines of business
- Support effective cross-selling
- Support mass customization

Identifying the business process

Identifying the business processes includes the following tasks:

- Understanding the existing business processes
- Identifying the business process changes
- Understanding the impact on related IT processes or areas

Identifying the information view

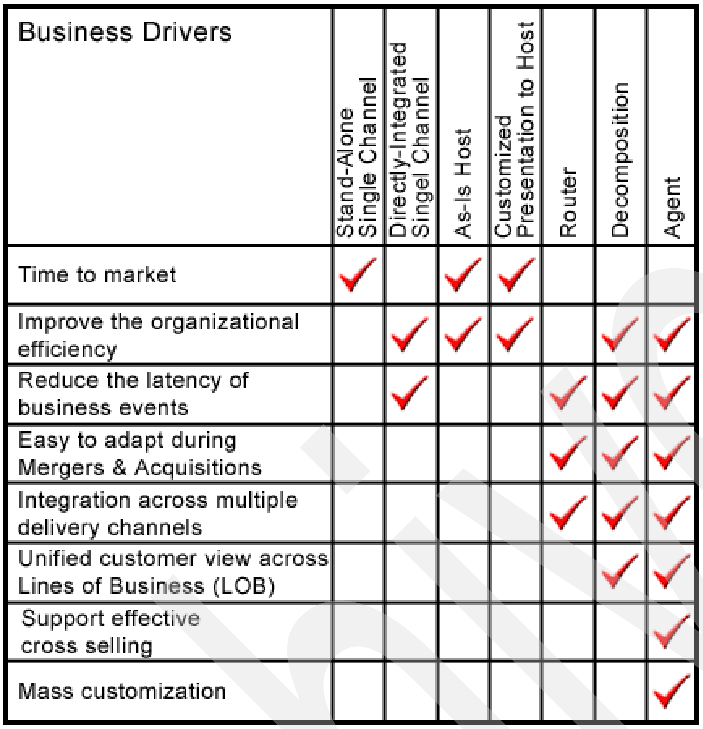
Finally, you need to list the aspects about the information view, for example:

- Minimize application complexity
- Improve scalability of the application
- Leverage existing skills
- Integrate with the backend applications
- Leverage the application applications
- Leverage existing investment
- Minimize TOC

2.3.2 Decision making tables

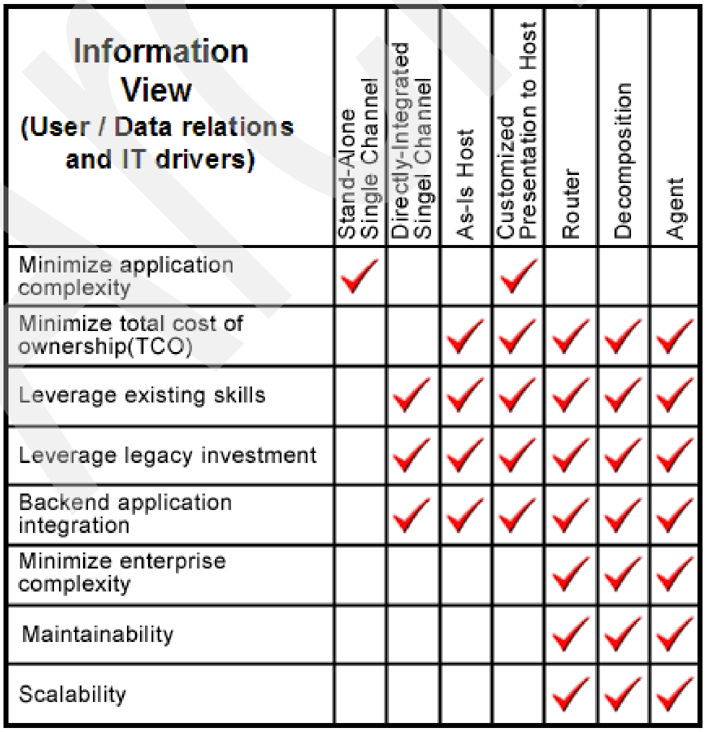
Based on this analysis and the answers that you obtain from completing the tasks, you need to map the answers to the items shown in Table 2-6 and Figure 2-7. This mapping will result in the application pattern that matches your business goals.

Table 2-6 Relationship between business drivers and self-service solutions types



Business Drivers	Stand-Alone Single Channel	Directly-Integrated Singel Channel	As-Is Host	Customized Presentation to Host	Router	Decomposition	Agent
Time to market	✓		✓	✓			
Improve the organizational efficiency		✓	✓	✓		✓	✓
Reduce the latency of business events		✓			✓	✓	✓
Easy to adapt during Mergers & Acquisitions					✓	✓	✓
Integration across multiple delivery channels					✓	✓	✓
Unified customer view across Lines of Business (LOB)						✓	✓
Support effective cross selling							✓
Mass customization							✓

Table 2-7 Relationship between the information view and self-service solutions types



Information View (User / Data relations and IT drivers)	Stand-Alone Single Channel	Directly-Integrated Singel Channel	As-Is Host	Customized Presentation to Host	Router	Decomposition	Agent
Minimize application complexity	✓			✓			
Minimize total cost of ownership(TCO)			✓	✓	✓	✓	✓
Leverage existing skills		✓	✓	✓	✓	✓	✓
Leverage legacy investment		✓	✓	✓	✓	✓	✓
Backend application integration		✓	✓	✓	✓	✓	✓
Minimize enterprise complexity					✓	✓	✓
Maintainability					✓	✓	✓
Scalability					✓	✓	✓

2.4 Understanding the requirements of the solutions

Understanding the requirements of the solutions is the second stage in the process of planning a new topology. The goal of this stage is to describe how the applications support the required functionality. Figure 2-10 highlights the main aspect that you must consider in this stage. This aspect is a functional view or solution requirements.

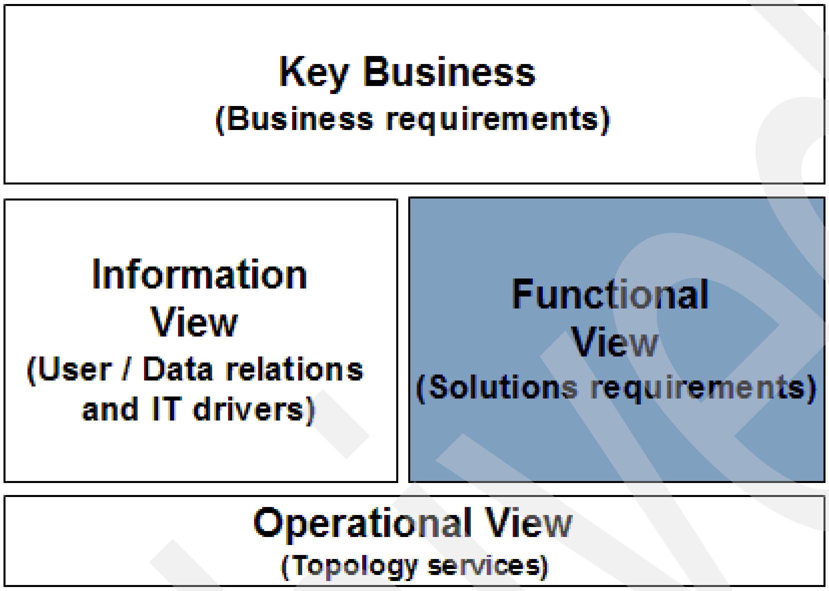


Figure 2-10 Planning for a new topology process: Main aspects for the second stage

2.4.1 Establishing the functional view or solution requirements

In this stage, you know what needs to be implemented by the solution. Now you need to establish a clear functional view. To achieve this goal, you must identify the following items.

1. Identify the solution components.
2. Identify how to build the components.
3. Identify where to place the components.

Identifying the solution components

The business requirements described in 2.3, “Mapping business requirements to the application patterns” on page 32, provide the input for the second stage. Now you need to translate them into functional requirements. The functional requirements are then translated into a high-level picture of functional components. Figure 2-11 shows an example for the Stand-alone Single Channel application type.

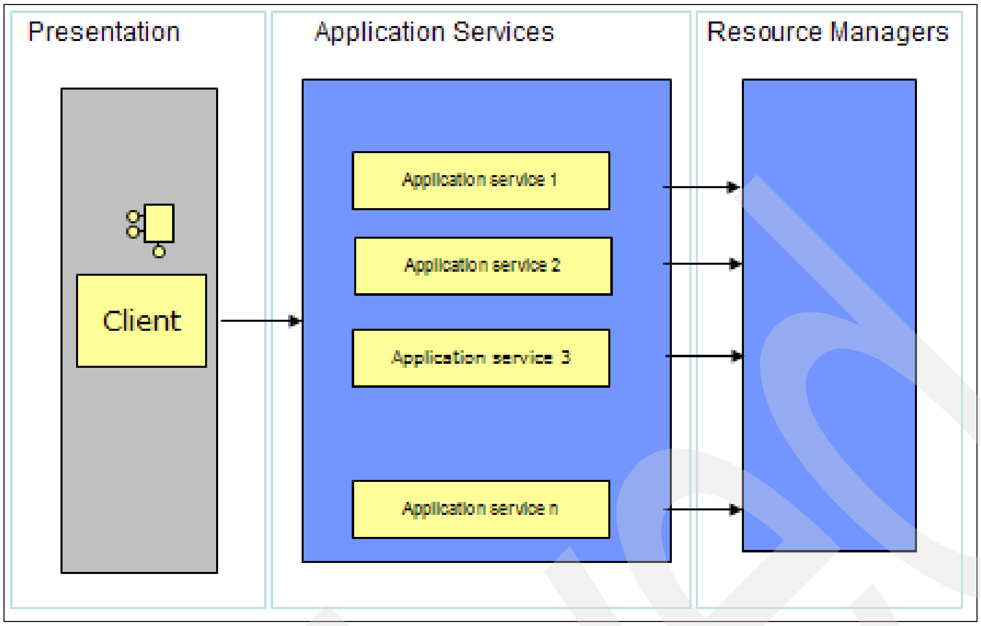


Figure 2-11 Layered architecture for the Stand-alone Single Channel application pattern

Identifying how to build the components

Next you select the way to build the functional components. Several proven approaches dominate the market. The most popular are J2EE and .Net architectures. Figure 2-12 illustrates how to build the functional components for the Stand-alone Single Channel application pattern according to the J2EE architecture.

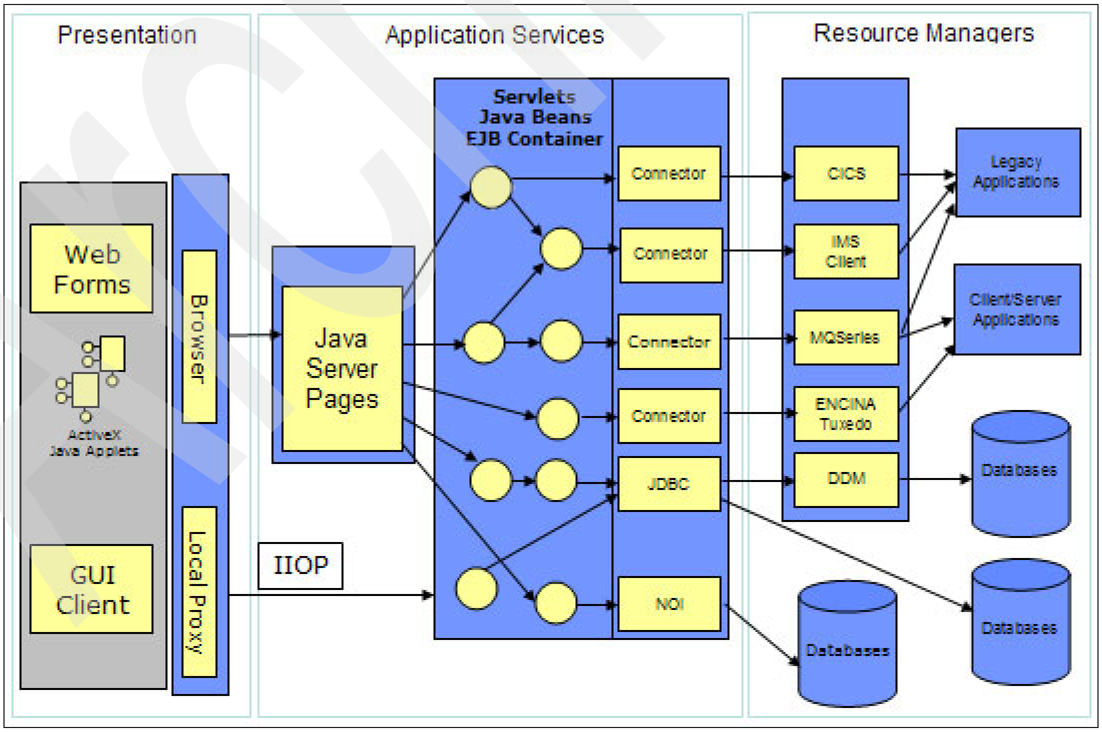


Figure 2-12 Identification of components according to the J2EE architecture

Identifying where to place the components

Finally, you need to translate the functional requirements into a high-level architecture of functional components. This layered architecture shows the relationship between each component. Figure 2-13 shows an example of the architecture of functional components for the Stand-alone Single Channel application pattern.

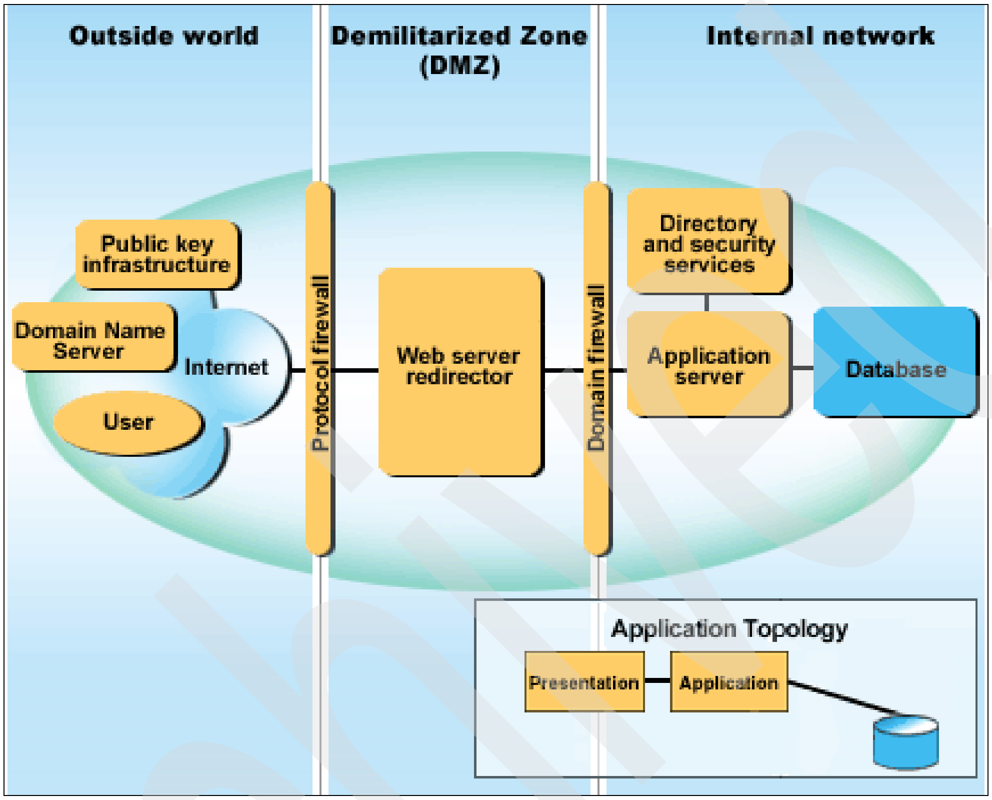


Figure 2-13 Relationship between the components of a Stand-alone Single Channel application pattern

Some functional components within this architecture type are:

- Public key infrastructure (PKI)

  The PKI component is a system for verifying the authenticity of each party involved in an Internet transaction, protecting against fraud or sabotage. This component is also for non-repudiation purposes to help consumers and retailers protect themselves against possible security threats.

- Domain Name Server (DNS)

  The DNS component assists in determining the physical network address associated with the symbolic address (Uniform Resource Locator (URL)) of the requested information.

- User

  The user component is most frequently a personal computing device (PC) supporting a commercial browser, such as Netscape Navigator and Internet Explorer.

- Protocol and domain firewall

  A firewall component is a system that manages the flow of information between the Internet and an organization’s private network. A firewall can separate two or more parts of a local network to control data exchange between these parts.

Firewalls control access from a less trusted network to a more trusted network. Traditional implementations of firewall services include screening routers (the protocol firewall) and application gateways (the domain firewall).

A pair of firewall nodes provides increasing levels of protection at the expense of increasing computing resource requirements. The domain firewall is typically implemented as a dedicated server node.

- Web application server

  To separate the Web server from the application server, a *Web server redirector* component is introduced. This component is used in conjunction with a Web server. The Web server serves HTTP pages, and the redirector forwards servlet and JavaServer Page (JSP) requests to the application servers. The advantage of using a redirector is that you can move the application server behind the domain firewall into the secure network, where it is more protected than within the demilitarized zone (DMZ).

- Application server

  The application server component provides the infrastructure for application logic and can be part of a Web application server. It is capable of running both presentation and business logic but generally does not serve HTTP requests. When used with a Web server redirector, the application server component can run both presentation and business logic. In other situations, it can be used for business logic only.

- Directory and security services

  The directory and security services component supplies information about the location, capabilities, and attributes of resources and users known to this Web application system. It can supply information for various security services (authentication and authorization) and can perform security processing, for example, to verify certificates. The authentication in most current designs validates the access to the Web application server part of the Web server, but this component also authenticates for access to the database server.

- Databases

  This component provides persistent data storage and retrieval in support of the user-to-business transactional interaction. The data stored is relevant to the specific business interaction, for example bank balance, insurance information, and current purchase by the user.

At this point, you should understand how applications will deliver the required functionality. Now you need to think about which topology will support the chosen solution.

2.5 Selecting the topology

The third stage in the process of planning a new topology is to select the topology. The goal of this stage is to define the infrastructure that is going to support the enterprise solutions. Figure 2-14 highlights the main aspect that you must consider in this stage. This aspect is the operational view or topology services.

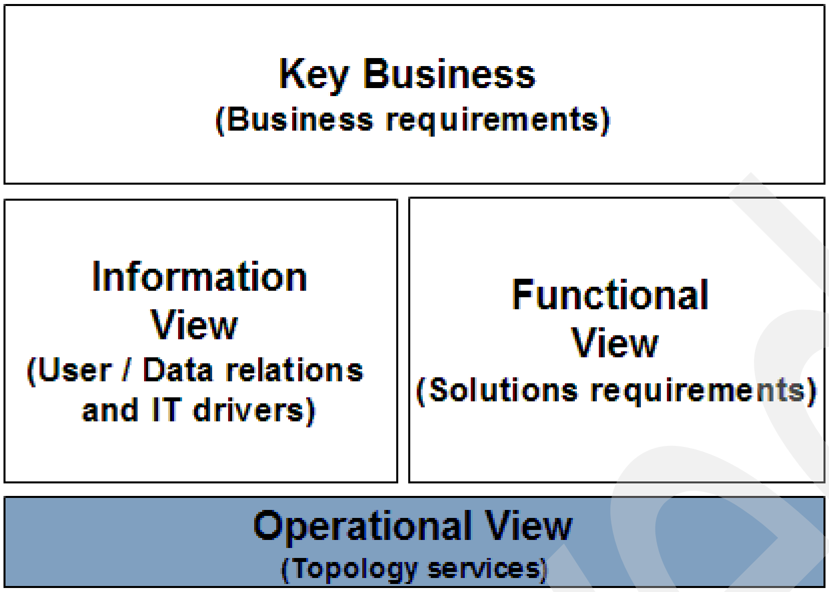


Figure 2-14 Planning for a new topology process: The main aspects for the third stage

When you select a new topology, several considerations can impact your decision of which topology to use. In this stage, you must describe the key concepts that will affect the new topology selection.

2.5.1 Establishing the operational view or topology services

Now that you know how the applications will support the required functionality, establish a clear operational view that support this selection. To achieve this goal, you must:

1. Identify nonfunctional requirements.
2. Identify the topology diagram.
3. Display the topology selection criteria.

Identifying nonfunctional requirements

The nonfunctional requirements for an e-business solution system address those aspects of the system that do not directly affect the functionality seen by users, but which can have a profound effect on the quality of service the solution provides. The nonfunctional aspects of an e-business solution system cover a broad range of operational areas such as:

- Performance
- Scalability
- Availability (including recoverability and reliability)
- Maintainability (including flexibility and portability)
- Security
- System Management
- Environment (including safety)
- System usability
- Data integrity (including currency, locality of updating and data retention)

Table 2-8 shows a sample of the nonfunctional requirements of an e-business solution.

Table 2-8 Sample nonfunctional requirements

Nonfunctional requirement	**Description**
Availability	► Provide services 24x7x365. ► Provide 99.25% overall site availability. ► Use redundant components. ► Use diverse routes. ► Use diverse access points. ► Use separate power sources for redundant components.
Backup and recovery	► Solution functionality should not be brought down during backups. ► The performance of the solution should not be impacted by more than 7% during backups. ► A complete restore of the complete solution should be completed within six hours. ► A complete restore of any server within the solution cluster should be completed within three hours.
Capacity	► Support 3500 peak concurrent customer requests. ► Support 17000 active customer sessions per peak period.
Performance	► Meet a 5-second home page download time with effective customer access of 56 Kbps. ► Meet a 5-second static-only content page download time with effective customer access of 56 Kbps. ► Meet an 8-second dynamic content page download time, including enterprise support system calls, with effective customer access of 56 Kbps.
Security	► Create separate security domains for Web presentation, applications, and secure data or enterprise assets. ► Provide for application and data access control within the application or a separate security application. ► Disable all non-critical services and ports within all servers and appliances. ► Restrict inbound and outbound traffic to only the required ports and addresses. ► Deny all traffic unless explicitly permitted to flow. ► Restrict Simple Network Management Protocol (SNMP) access to designated devices and paths. ► Disable all Web-based management interfaces.
Scalability	► Develop the solution to allow all components to scale at least three times the expected user levels.
System management	► Use infrastructure components that manage using industry-standard SNMP. ► Provide a separate and secure management network for control, reporting, and monitoring. ► Provide for management consoles or applications to monitor operations and report on capabilities.

For more details about the nonfunctional requirements, see *WebSphere Application Server V6: Planning and Design WebSphere Handbook*, SG24-6446.

Identifying the topology diagrams

We have now selected the main nonfunctional requirements that best meets the needs of our e-business solution type. This is high availability (HA). The other aspects are considered in the topology but as not main aspects.

The basic principals of HA are simple. You must consider the possible point of failure of each component on your solution. This will impact final decisions and such aspects as technology, topology, and products used to build the overall solution.

Figure 2-15 shows an example of the topology diagram for HA.

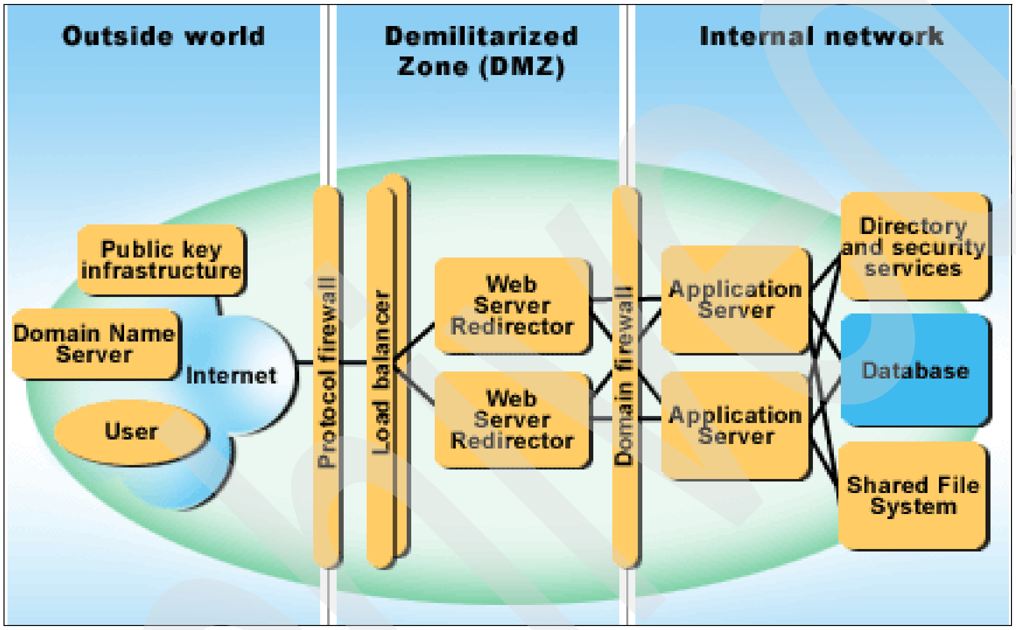


Figure 2-15 Topology diagram: Separation of components for high availability

This topology splits the Web Application Server node into two functional nodes by separating HTTP server function from the application server function. The Web server redirector node decides whether the request is being served by the local HTTP server or forwarded to the application server nodes. The application server nodes are behind the domain firewall adding security.

Topology summary

Table 2-9 summarizes the topology selected in our example.

Table 2-9 Topology summary

Component	Requirement	Description
Presentation tier ► Load balancer ► Web server ► Protocol and domain firewall	High availability, performance and security	► High availability – Primary/backup for load balancer – Active/active for Web server – Primary/backup for protocol firewall and domain firewall ► Performance – Load balancer – Caching proxy – Clustering ► Security – HTTPS – Protocol firewall (outside) – Domain firewall (inside)
Application tier ► Application server ► Directory service		► High availability – Primary/backup for LDAP component – Active/active for application server ► Performance – Dynamic caching – Clustering ► Security – Separate the Web server and load balancer into a DMZ
Backend tier: ► Database server		► High availability – Primary and backup configuration ► Performance – Provided for the software solution ► Security – Not covered

Advantages and disadvantages of the selected topology

It’s important to understand the advantages and disadvantages of a selected topology to make a decision if the selected topology makes a business sense. For example, you should estimate if the money you save with the selected topology (such as reducing down time) outweighs the investment into building the selected topology.

Here are the benefits of the topology in our example:

- Each component can be configured to the specific tasks.
- Each caching proxy component adds scalability, increased bandwidth, and failover capabilities to the caching and security configuration.
- The load balancer nodes provide automatic backup and recovery from failures within both the load balancer component and the caching proxy component.
- It’s easy to add and remove additional caching proxy nodes as network traffic demands.
- A domain firewall between the two node types adds more security.

The disadvantages of this topology are:

- Additional hardware is needed to implement this topology.
- Many configuration changes in the application server node require redeployment of the plug-in-cfg.xml file to all Web server redirector nodes.
- The configuration of the domain firewall requires additional skills.

2.6 Reviewing the products mapping

After you select the topology, you must map the functional components defined in the topology diagram to specific products which implement the e-business solution on a selected platform. The product mapping identifies the platform, software product name, and often version numbers. Table 2-10 shows an example of such a review.

Table 2-10 Reviewing the product mapping

Component	Product
Firewall	Hardware: Two Cisco PIX 506E firewalls Software: Cisco PIX Software Version 7.0
Load balance server	Hardware: Two IBM ThinkCentre S Series systems Software: Windows 2000 server with SP4
Web server redirector	Hardware: Two LPARs running on iSeries model 520 with 512 MB of RAM Software: IBM HTTP v6, IBM HTTP Plug-in v6, i5/OS V5R3
Application server	Hardware: Two LPAR running on iSeries model 520 with 2 GB of RAM Software: IBM WebSphere Application Server v6 Network Deployment, i5/OS V5R3
Directory server	LDAP service in i5/OS V5R3
Database server	DB2 Universal Database (UDB) for iSeries V5R3

Table 2-11 shows some of the most important aspects to consider when deciding on a platform to host an e-business application.

Table 2-11 Aspects to consider in the product selection stage

Aspect	Description
Support platforms	One of the first steps in any hardware platform selection should be to determine which operating system platforms are supported by the software components of the solution.
Customer IT standards and skill	Use your IT standards to help reduce the choices in hardware platform.
Existing topology decisions	Review this document for any server, operating system, and storage platforms decisions that may have already been made. For example, a decision may already be documented to use an AIX® platform for all servers and Windows platforms only when a software component cannot run under AIX.
Existing nonfunctional requirements	Review any existing nonfunctional requirements to understand the possible impact on server and storage selection. For example, for administration simplification requirements, you may select an iSeries server as the best fit.
Pricing impact	The price of the hardware platform is frequently one of the most visible components of a solution and should be considered.

3

Topology implementation guidelines

This chapter is positioned between Chapter 2, “Planning for a new topology” on page 21, and Part 2, “Implementation steps” on page 59, to prepare you for the implementation steps. The implementation of a distributed, highly available topology is a sophisticated issue that requires much consideration. The more time you spend thinking about all details in advance, the better result you will achieve by spending less time on the implementation stage.

This chapter was written under the assumption that you have already completed the planning session as explained in the previous chapter. As a result of completing that session, you should know the final topology concept that will be implemented. In this chapter, you work with this chosen topology and review the important guidelines to assist you in preparing for the implementation phase.

You can compare the concepts presented in this chapter to cooking. For example, the planning session is similar to choosing a recipe that you want to prepare. First you gather all the necessary ingredients (components) and become familiar with how to prepare the recipe (implementation). When you start cooking (implementation phase), you have all your ingredients ready and add them in the proper sequence. If you miss an ingredient or skip a cooking step, your meal may have unexpected results. Likewise, missing a component, or skipping an implementation step, can cause problems with your final topology.

3.1 Planning phase output

The planning phase, as explained in Chapter 2, “Planning for a new topology” on page 21, is one of the first tasks that you must complete. All of your staff that is involved in the planning session must understand the company’s business view, functional view, and operational view. Based on this information, you can plan the topology.

In addition, you must understand the company’s preferences and try to include them in the topology. For example, consider the type of hardware that will be used to implement the topology. The result of the planning session is to have a particular topology that you need to implement. Figure 3-1 shows the sample topology that is used throughout this chapter.

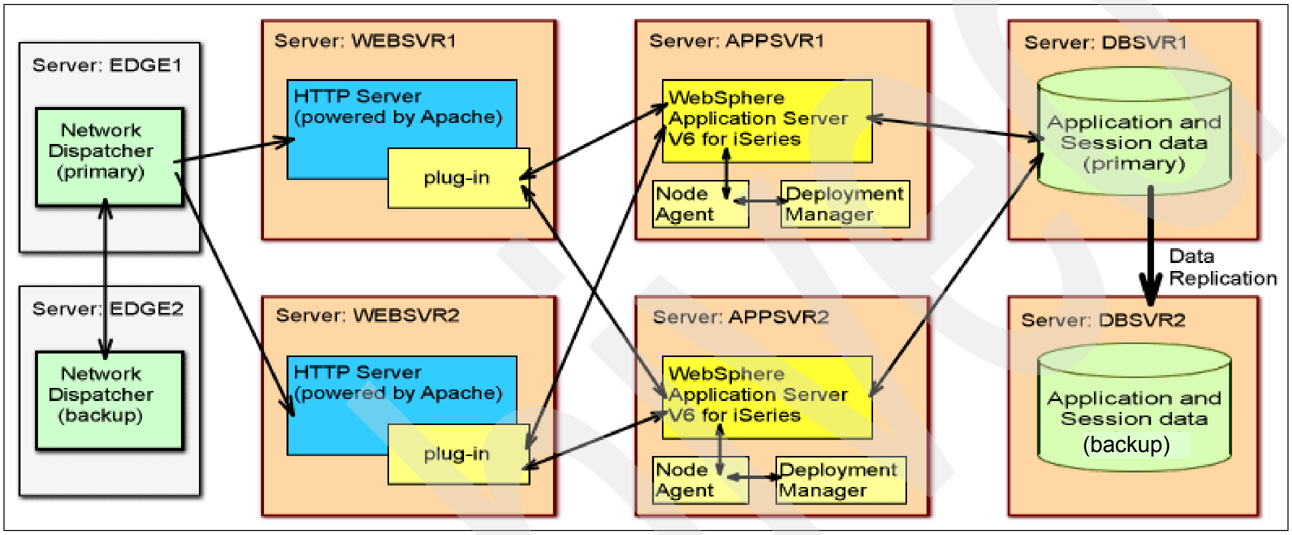

Server: EDGE1
Network Dispatcher (primary)
Server: EDGE2
Network Dispatcher (backup)
Server: WEBSVR1
HTTP Server (powered by Apache)
plug-in
Server: WEBSVR2
HTTP Server (powered by Apache)
plug-in
Server: APPSVR1
WebSphere Application Server V6 for iSeries
Node Agent
Deployment Manager
Server: APPSVR2
WebSphere Application Server V6 for iSeries
Node Agent
Deployment Manager
Server: DBSVR1
Application and Session data (primary)
Data Replication
Server: DBSVR2
Application and Session data (backup)

Figure 3-1 Large customer sample topology

Before you start the implementation phase, you must have a complete understanding of the final topology. In the case of complex topologies, you should know how each component is involved and how it interacts with other components. Close contact and cooperation with the planning team is an important factor during the implementation phase. Try to use all the information that has already been agreed upon during the planning session and avoid trying to “reinvent the wheel” or do what has already been done before.

The goal of this chapter is to prepare you for the implementation phase and to make the implementation tasks as easy as possible. Before delving into this chapter, you must have all the information you need and contact the planning team if you require more details to implement the topology.

Table 3-1 shows general information regarding the servers in this topology. It contains the list of servers together with the hardware and operating system used. If any of the servers are divided into logical partitions (LPARs), you must also include this information.

Table 3-1 Physical servers inventory

Server name	Hardware platform	Operating system	Function
WEBSVR1	@server i5	i5/OS V5R3	Web server/cell 1
WEBSVR2	@server i5	i5/OS V5R3	Web server/cell 2
APPSVR1	@server i5	i5/OS V5R3	Application server/cell 1

Server name	Hardware platform	Operating system	Function
APPSVR2	@server i5	i5/OS V5R3	Application server/cell 2
DBSVR1	@server i5	i5/OS V5R3	Primary database server
DBSVR2	@server i5	i5/OS V5R3	Backup database server
EDGE1	IBM @server xSeries®	Windows 2000 Server	Primary load balancer
EDGE2	xSeries	Windows 2000 Server	Backup load balancer

Table 3-2 shows information for defining the network settings. At this step, you should know the IP addresses, subnet masks, and physical network adapters that will be assigned to the servers. If some of the servers have more than one network adapter, the table should also include relationships between network interface controller (NIC) and the Transmission Control Protocol/Internet Protocol (TCP/IP) address.

Table 3-2 Network inventory

LPAR name or server name	Network adapter or line description	Network
WEBSVR1	EHTLINE	DMZNET
WEBSVR2	EHTLINE	DMZNET
APPSVR1	EHTLINE	CORPNET
APPSVR2	EHTLINE	CORPNET
DBSVR1	EHTLINE	CORPNET
DBSVR2	EHTLINE	CORPNET
EDGE1	Ethernet adapter	DMZNET
EDGE2	Ethernet adapter	DMZNET

Table 3-3 contains information about the software that is installed on each server.

Table 3-3 Software inventory

LPAR or server name	Software installed
APPSVR1	WebSphere Application Server V6 Network Deployment
APPSVR2	WebSphere Application Server V6 Network Deployment
WEBSVR1	IBM HTTP Server (powered by Apache)
WEBSVR2	IBM HTTP Server (powered by Apache)
EDGE1	Edge Components for IBM WebSphere Application Server ND V6.0
EDGE2	Edge Components for IBM WebSphere Application Server ND V6.0

Important: Some licensed program products (LPPs) are licensed based on the quantity of users, while others are licensed based on the number of the processors in a server or LPAR. Be sure to consider all licensing information.

After you collect all the information about the environment, you are almost ready to start. One of the last points to determine is who will set up this complex topology, which covers so many different components. In making this decision, you must consider the skills that are needed to implement the topology.

Based on the chosen topology, isolate all the components that comprise the topology. For each component, prepare a list of required skills. This list will allow you to match the required skills to those that already exist in your company and help you to determine which skills are missing.

To fill the gap of the missing skills, decide whether you want to schedule training for your existing employees or to outsource the work. The option that you select will depend on how much time is available before you start the implementation phase. Whether you choose to train your employees or outsource the work will also depend on the costs involved. You must also consider whether the investment in these skills for this project will be beneficial to you in any future projects. If the missing skills are required only for implementing one particular topology, then consider outsourcing the work. However, if the skills are required now and in the future, then consider training your employees.

Consider the following categories and related questions:

- Current skills
  - Do you have an accurate skills list?
  - Do you have job specifications for the current skills?
- Required skills
  - What skills are required for the new task?
- Critical skills
  - Do all of these skills exist?
  - Do you have a job specification for all critical skills?
- Skills retention
  - Are existing resources encouraged to maintain their skills?
  - Is there a skills development plan for existing resources?
  - Are you likely to lose resources as result of the project?
  - Are these critical skills?

LPAR considerations

Be aware that some LPAR configurations force the purchase of additional hardware such as disk controllers and Ethernet cards. To configure the environment from a hardware perspective, use a dedicated tool that assists with determining the required hardware. You can use the LPAR Validation Tool (LVT) to validate partitioned servers. To learn more about the LVT tool, see the following Web site.

`http://www.ibm.com/servers/eserver/iseries/lpar/systemdesign.htm`

3.2 Installation and configuration guidelines

This section helps you to prepare for the implementation stage. It covers all the details that you must consider prior to implementation, as well as the components which comprise the topology. Specifically, it covers these components:

- Edge Server
- HTTP Server
- WebSphere Application Server
- Database server
- Network infrastructure

3.2.1 Edge Server

Edge Components (formerly Edge Server) are now a part of the WebSphere Application Server offering. You can use Edge Components in conjunction with WebSphere Application Server to control client access to Web servers. You can also use them to enable enterprises to provide better service to users who access Web-based content over the Internet or a corporate intranet. Edge Components help to reduce Web server congestion, increase content availability, and improve Web server performance. As the name indicates, Edge Components are usually installed on machines that are close (in a network configuration sense) to the boundary between an enterprise's intranet and the Internet.

The WebSphere Application Server ND V6.0 Edge Components includes the Caching Proxy and Load Balancer components.

Caching Proxy

Caching Proxy reduces bandwidth use and improves a Web site's speed and reliability by providing a point-of-presence node for one or more back-end content servers. Caching Proxy can cache and serve static and dynamic content generated by WebSphere Application Server.

Network performance is affected by the introduction of Caching Proxy functionality. Use Caching Proxy alone or in conjunction with Load Balancer to improve the performance of your network. Figure 3-2 shows an example of a topology and the placement of Caching Proxy in a network.

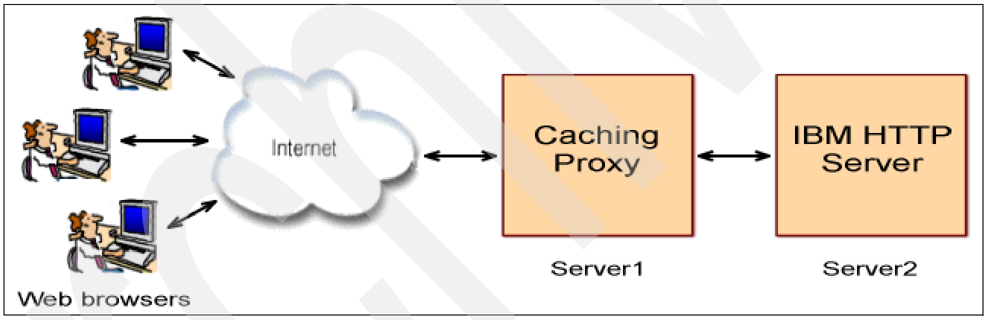


Figure 3-2 Caching Proxy demonstration network

To build a Caching Proxy network, as shown in Figure 3-2, review the following high level instructions.

Required computer systems and software

You need the following computer systems and software components:

- A computer system to act as Server1: This system must have access to the Internet.
- A computer system to act as Server2: An HTTP server must be installed on this system.
- A computer system to act as the workstation: A Web browser must be installed.

Building Server1

Install and configure the Caching Proxy as follows:

1. Ensure that the computer server meets all hardware and software requirements.
2. Install the Caching Proxy component as a super user (typically root).

Configuring the server

From the workstation, use these steps:

1. Start a Web browser.
2. In the Address field of your browser, type `http://server1`, where *server1* refers to the host name or IP address of the machine designated as Server1.
3. Click **Configuration and Administration Forms**.
4. Complete all steps as explained in the configuration manual (comes with the product).
5. Click **Submit**.
6. Restart the server.

Testing the Caching Proxy network

From the workstation, complete these tasks:

1. Start a Web browser.
2. In the Address field of the browser, type `http://server1`.

   HTML pages from Server2 are then proxied through Server1 and delivered to the Web browser.
3. To access the Configuration and Administration forms, in the Address field of the browser, type `http://server1/pub/`.

The home page of the Configuration and Administration forms is displayed.

> **Important:** For more details about Caching Proxy, see the WebSphere InfoCenter:
>
> `http://www.ibm.com/software/webservers/appserv/doc/v60/ec/infocenter/index.html`

Load Balancer

Load Balancer, also known as Network Dispatcher, is an edge-of-network system that directs network traffic flow, reduces congestion, and balances the load of various services and systems. Load Balancer provides site selection, workload management, session affinity, and transparent failover. This section discusses the network hardware issues to consider when introducing Load Balancer functionality into the network.

The topology shown in Figure 3-3 is used in considering a Load Balancer.

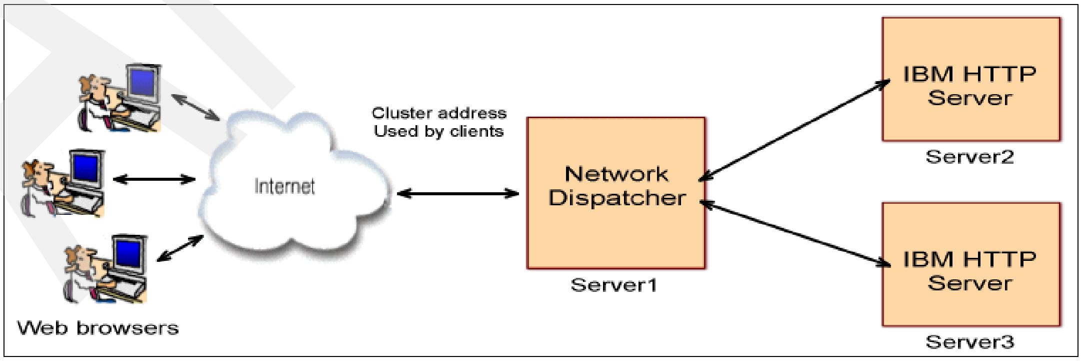


Figure 3-3 Load Balancer demonstration network

To build a Load Balancer network, review the high level instructions presented in the following sections.

Required computer systems and software

You need the following computer systems and software components:

- Computer system to act as the Dispatcher (Load Balancer)

  This system requires two IP addresses: one for the regular host address of the server and a second address required for load-balancing purposes. The second address, also known as a *cluster address*, is associated with your company Web site and is known to the DNS server.

- Two computer systems to act as Web servers

  Each HTTP server requires two IP addresses: a regular IP address and a *VIRTUALIP address. The *VIRTUALIP address must match the cluster address used by Load Balancer. Both addresses also need to be bound to the applicable HTTP server instance on each HTTP server system in the cluster. If Message Authentication Code (MAC)-based forwarding is used, then the Load Balancer cluster address and the address used by each HTTP server in the cluster must be in the same subnet.

Configuring the network

Follow these high-level steps to set up this configuration:

1. Set up the workstations so that they are on the same local area network (LAN) segment. Ensure that network traffic among the three machines does not pass through any routers or bridges.
2. Configure the network adapters on all workstations.
3. Ensure that each workstation contains only one standard Ethernet NIC.
4. Ensure that `server1.company.com` can ping both `server2.company.com` and `server3.company.com`.
5. Ensure that `server2.company.com` and `server3.company.com` can ping `server1.company.com`.
6. Ensure that content is identical on the two Web servers (Server2 and Server3).
7. Ensure that the Web servers on Server2 and Server3 are operational. Use a Web browser to request pages directly from both `http://server2.company.com` and `http://server3.company.com`.
8. Obtain another valid IP address for this LAN segment. This is the address that you provide to clients who want to access your site. It is also the cluster IP address used by Load Balancer.
9. Configure two IP addresses on each of the HTTP server systems. One is a regular IP address that must match what is used to define the HTTP server system within the Load Balancer software. The second address must be a *VIRTUALIP address that needs to match the cluster address defined in Load Balancer.
10. Ensure that the HTTP server instance on each Web server system is bound to both of the addresses configured in the previous step.

Configuring Load Balancer

With Load Balancer, you can create a configuration by using the command line, the configuration wizard, or the graphical user interface (GUI). If the configuration wizard or GUI is used to configure Load Balancer, then ensure that the following directive exists in the configuration file that is generated:

```
dscontrol executor configure www.company.com
```

If this directive does not exist, add that line toward the bottom of the file, after the Web server declarations.

Testing the Load Balancer network

To finalize the setup of the Load Balancer, test it using the following instructions.

1. From a Web browser, go to the following address to verify that a page appears.

   ```
   http://www.company.com
   ```

2. Reload the page in the Web browser.
3. Type the following command:

   ```
   dscontrol server report www.company.com:80:
   ```

4. Verify that the Total connections column of the two servers adds up to two.

Important: For the most current information about hardware and software requirements, refer to the WebSphere Application Server Supported hardware and software Web site.

http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html

You must also consider the following network hardware issues.

- Memory considerations

  A large amount of memory must be dedicated to the proxy server. Caching Proxy can consume 2 GB of virtual address space when a large memory-only cache is configured. Memory is also needed for the kernel, shared libraries, and network buffers. Therefore, it is possible to have a proxy server that consumes 3 or 4 GB of physical memory.

- Hard disk considerations

  It is important to have a large amount of disk space on the machine on which Caching Proxy is installed. This is especially true when disk caches are used. Reading and writing to a hard disk is an intensive process for a computer.

- Network considerations

  Network requirements, such as the speed, type, number of NICs, and the speed of the network connection, to the proxy server, affect the performance of Caching Proxy. It is generally in the best interest of performance to use two NICs on a proxy server machine: one for incoming traffic and one for outgoing traffic. Furthermore, NICs should be at least 100 MB, and they should always be configured for full-duplex operation. Finally, the speed of the network connection is also important.

- Central processing unit (CPU) considerations

  The CPU of a Caching Proxy machine can become a bottleneck factor. CPU power affects the amount of time it takes to process requests and the number of CPUs in the network affects scalability. It is important to match the CPU requirements of the proxy server to the environment, especially to model the peak request load that the proxy server will sustain.

3.2.2 HTTP Server

If Web serving is a critical aspect of your business, you may want high availability (HA) and scalability of your Web server environment. High availability and scalability of the Web server environment can be achieved through the use of iSeries clustering. You can learn more about iSeries clustering in the iSeries Information Center.

http://publib.boulder.ibm.com/infocenter/iseries/v5r3/index.jsp?lang=en

You should setup HTTP Server cluster especially if you need to maintain high available Common Gateway Interface (CGI) applications. High availability of CGI programs is achieved through the clustered hash table, which acts as a focal point for the sessions. However in this redbook, we focus on the applications developed according to the Java 2 Platform, Enterprise

Edition (J2EE), standard instead of CGI programs. J2EE applications that run on the WebSphere Application Server manage session state through the Session Manager, which is a part of the application server. This is why we do not have to set up session management at the level of the HTTP Server. Finally, it is not necessary to set up an HTTP Server cluster based on iSeries clustering.

In this section, we focus on the peer HTTP model that assumes that high availability is assured through the Edge Components. Figure 3-4 shows a sample peer scenario.

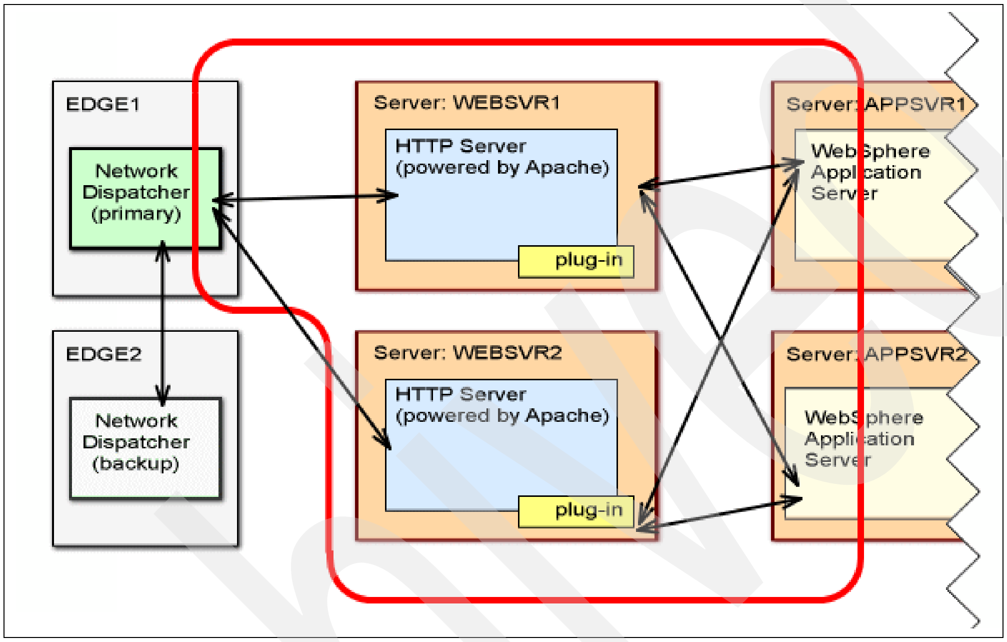

EDGE1
Network Dispatcher (primary)
EDGE2
Network Dispatcher (backup)
Server: WEBSVR1
HTTP Server (powered by Apache)
plug-in
Server: WEBSVR2
HTTP Server (powered by Apache)
plug-in
Server: APPSVR1
WebSphere Application Server
Server: APPSVR2
WebSphere Application Server

Figure 3-4 HTTP Server peer topology

In this model, there is no declared primary node. All nodes are in an active state and serve client requests. Network Dispatcher, for example the IBM WebSphere Edge Server, evenly distributes requests to different cluster nodes. This guarantees distribution of resources in cases of heavy client loads.

Review Table 3-4 for the mandatory and optional software used by the HTTP Server (powered by Apache).

Table 3-4 HTTP Server prerequisites

Product name	**Product number**	**Comment**
IBM HTTP Server for iSeries	5722-DG1	The LPP is IBM HTTP Server for iSeries.
TCP/IP Utilities	5722-TC1	This is a useful collection of TCP/IP applications including Telnet and File Transfer Protocol (FTP).
Java Developer Kit 1.3	5722-JV1 *Base Option 5	Your HTTP Server (powered by Apache) requires this LPP for the administration GUI, commonly referred to as the admin GUI.
HA Switchable Resources	5722-SS1 Option 41	Optional: Use this option if you plan to implement highly available Web server solutions for CGI applications.

Product name	Product number	Comment
Digital Certificate Manager (DCM)	5722-SS1 Option 34	Optional: This is used to support the handling of digital certificates used by Secure Sockets Layer (SSL) and Transport Layer Security (TLS) for secure Web serving
Cryptographic Access Provider	5722-AC3	Optional: If you want to use SSL or TLS, you must install one of the IBM Cryptographic Access Provider products.

3.2.3 WebSphere Application Server

In this section, you learn the steps that are required to install and create a basic configuration of WebSphere Application Server for iSeries. In this example we use a dual cell topology as shown in Figure 3-5. A primary advantage in using a dual cell topology is the fact that one cell can be taken offline for applications or system upgrades and the enterprise is still functional.

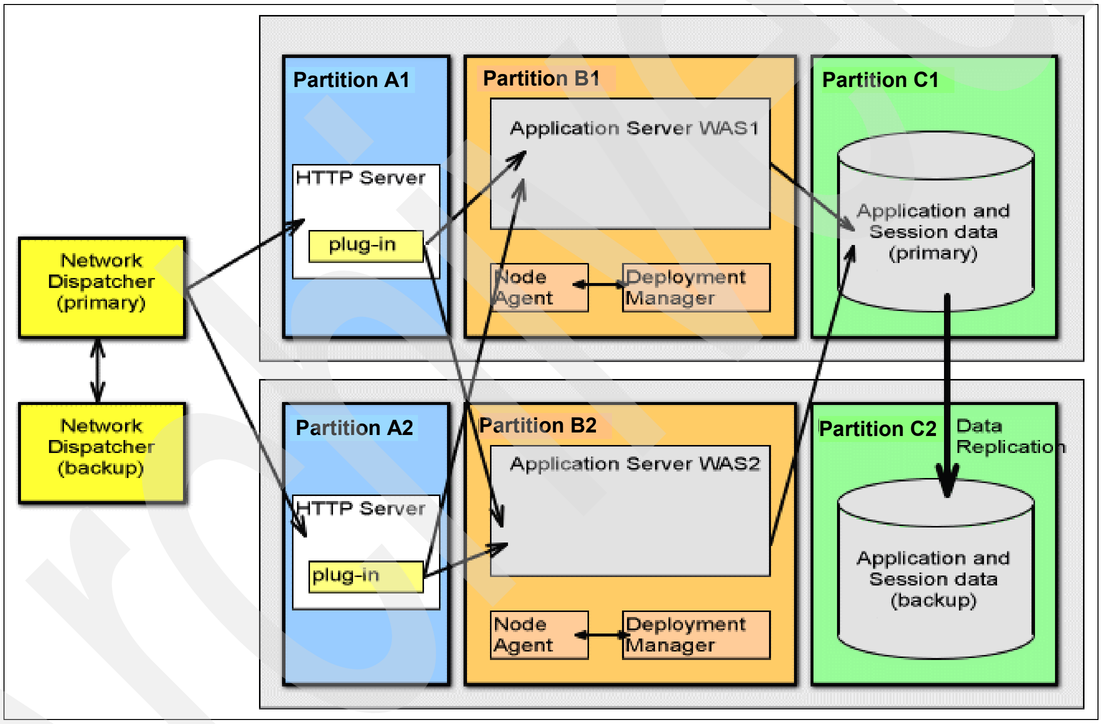


Figure 3-5 Dual cell WebSphere Application Server topology

Note: In this topology, we show only one application server profile per cell. This is done to simplify the diagram. In a production environment, you would configure multiple application server profiles for high availability and scalability.

The user who performs this installation doesn't have to be a WebSphere Application Server expert, but needs basic iSeries operator skills such as loading and applying program temporary fixes (PTFs) and installing LPPs. In addition, the OS/400 user profile must have *ALLOBJ and *IOSYSCFG special authorities to implement the steps indicated in the following list.

- When planning the installation of WebSphere Application Server, consider these items:
  a. Ensure that the iSeries meets all prerequisites for the software and hardware system.
  b. We recommend that you use a specific OS/400 Cumulative PTF Package for a WebSphere Application Server environment on iSeries. Ensure that the latest package is applied on the iSeries.
  c. Order the latest level of the WebSphere Application Server Group PTF. This PTF upgrades WebSphere Application Server to the latest FixPak level and applies the Group PTFs for the IBM Developer Kit for Java, IBM DB2 Universal Database (UDB), and IBM HTTP Server.
  d. Read and understand the release notes for the WebSphere Application Server for iSeries Group PTF.
- Before and after you install WebSphere Application Server, consider these items:
  a. Install an OS/400 Cumulative PTF package at this time if the iSeries system requires it.
  b. You can install WebSphere Application Server for iSeries from a LAN-attached workstation. However this checklist assumes the use of the iSeries CD-ROM drive.
  c. Ensure that the iSeries system is *not* in restricted state.
  d. Start the installation script from a Qshell session. The installation may take up to two hours to complete.
  e. Install the WebSphere Application Server Group PTF according to the cover letter instructions. Select to IPL the system after the PTFs are loaded and applied.

  > **Note:** For details about installing WebSphere Application Server V6 on iSeries, refer to Appendix B, “Installing WebSphere Application Server Version 6” on page 255.

  f. Before you start the WebSphere Application Server environment, set the usage limit from the WebSphere Application Server for iSeries Proof of Entitlement (POE) or invoice.
  g. Verify that the TCP/IP interface on the iSeries system’s Internet address and the LOOPBACK interface are active.
  h. Make sure that TCP/IP host and domain names are defined on the iSeries system. A value must exist for both the host name and the domain name parameters because the host name should not have a value of *NONE.
  i. Ensure that the host name from the previous step exists in the iSeries system’s host table.
- When you configure WebSphere Application Server, consider these items:
  a. Create an unmanaged Application Server profile in each cell.
  b. Create a Network Deployment profile in each cell.
  c. Federate an application server profile to the corresponding cell.
  d. Create horizontal clusters.
  e. Configure resources.
  f. Install applications. For test purposes, we suggest using the Trade6 application.
  g. Enable session persistence.
  h. Generate a plug-in file.

3.2.4 Database server

To implement the WebSphere dual cell topology as shown in Figure 3-5 on page 54, you must consider access to enterprise data, which is critical for most e-business applications. Since many of these applications must be available around the clock, the associated data must also be highly available. In this section, we explain what you can do to make data highly available and the use of independent auxiliary storage pools (IASPs), as shown in Figure 3-6.

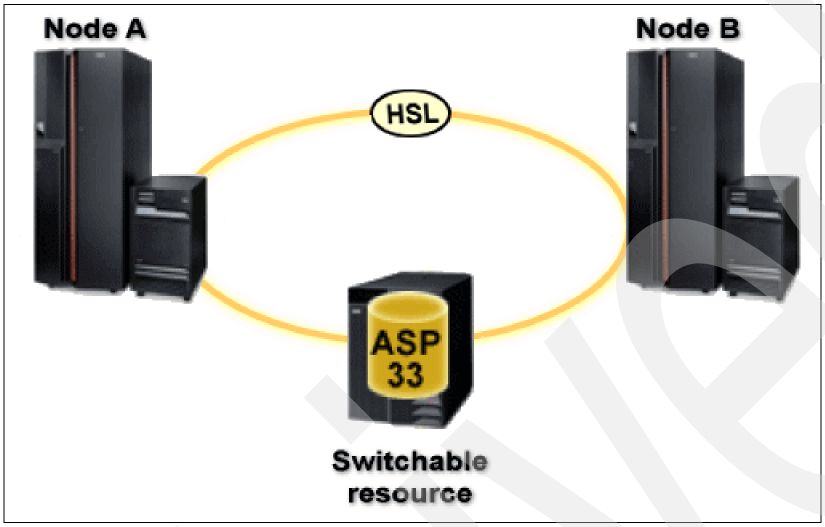


Figure 3-6 Switched disks using IASP

You use the following high-level steps to configure a switched disk environment for a remote database tier that can be used by a WebSphere test application.

1. Plan for system hardware and software requirements.
2. Cable a high-speed link (HSL) loop that contains the switched disk tower and the two nodes. These components comprise the cluster.
3. Use iSeries Navigator make changes to interfaces and I/O resource ownership as required.
4. Use Management Central options from iSeries Navigator to create cluster using two secondary partitions.
5. Using iSeries Navigator, enable the disk tower to be switchable.
6. Using iSeries Navigator, create a switchable hardware group with a new disk pool (IASP).
7. Migrate the enterprise database collection to the newly created IASP.
8. Create a library in the IASP to be used for persistence of WebSphere servlet session data.
9. Modify or create applicable Java Database Connectivity (JDBC™) data source definitions, using the WebSphere Administrative Console, to allow access to both the enterprise database and WebSphere session data collections, that are stored within the IASP.

   Usage of a takeover IP address associated with the database server is required to achieve high-availability characteristics when failover for the database tier occurs. Also, we highly recommend that you explicitly journal access paths or use a low system-managed access path protection (SMAPP) setting via the EDTRCYAP CL command. This allows all the database files in the disk pool to be available immediately to your application after the takeover IP address becomes active at the end of failover processing.
10. Perform switchover and failover tests to validate that the final IASP configuration provides the expected high-availability characteristics.

Cross-site mirroring

With cross-site mirroring (XSM), you can achieve disaster recovery protection since there is a second copy of your data that can be stored in a server at a separate geographic location. The nodes participating in geographic mirroring must be in the same cluster, the same device domain, and the same cluster resource group.

You need to install OS/400 - HA Switchable Resources licensed software (5722-SS1 Option 41). Option 41 provides the capability to switch independent disk pools between systems. Option 41 also enables you to use the iSeries Navigator cluster management interface to define and manage a cluster that uses switchable resources.

Configuring geographic mirroring with switchable independent disk pools

To configure geographic mirroring, you must first create the independent disk pool that will be mirrored. Before using iSeries Navigator, define up to four, one-to-one, data port TCP/IP routes bidirectionally as part of the connection between all the nodes in the cluster resource group. Geographic mirroring allows you to maintain an exact copy of the independent disk pool on a system at a different location for both protection and availability purposes. Configuring the independent disk pool to be switchable between nodes, at the same site in the cluster, allows for greater availability options.

Figure 3-7 shows an example of geographic mirroring between sites and both sites using switchable independent disk pools.

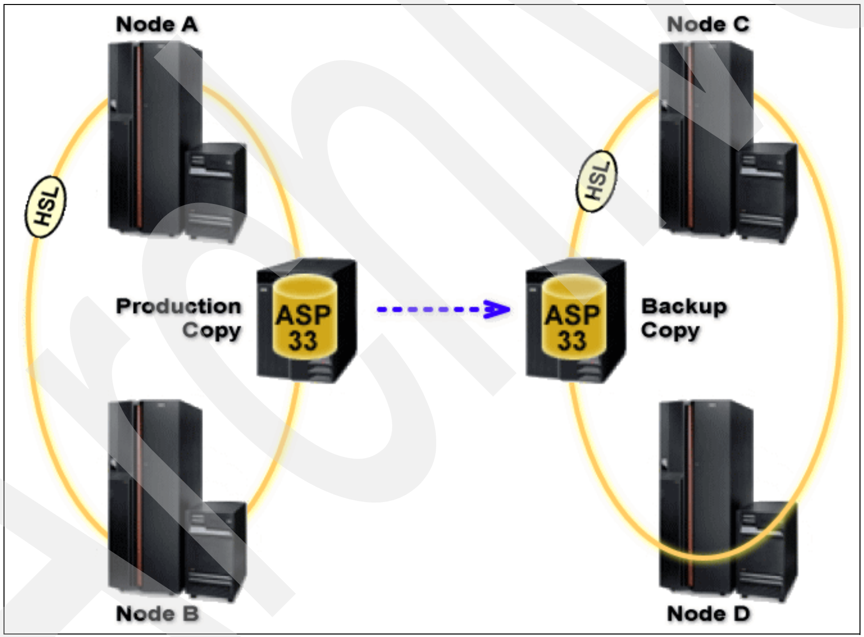


Figure 3-7 Geographic mirroring with switchable independent disk pools

To configure geographic mirroring with switchable independent disk pools, review the following high-level steps.

1. Plan and configure your TCP/IP routes for the data port.
2. Create a cluster containing all four nodes.
3. Make the hardware switchable, for switching between nodes A and B, and between nodes C and D.
4. Create a switchable hardware group.
5. Create the IASP on the primary node.

6. Define geographic mirroring sites.
7. Configure geographic mirroring.
8. Print and store your disk configuration.

Important: You can find more information about setting up geographic mirroring with switchable independent disk pools in 5.3, “Data considerations for high availability” on page 99, and in the iSeries Information Center at:

http://publib.boulder.ibm.com/infocenter/iseries/v5r3/index.jsp?lang=en

There are also several partners’ solutions in support of DB2 UDB for iSeries replication.

3.2.5 Network infrastructure

One of the major items to consider in availability is the network itself. When planning for a network, you must address capacity and accessibility as they are planned for the system itself. When major hubs or routers are down, users have difficulty accessing key business applications.

Ultimately, the network must be stable and easily recoverable to support core business applications. Employ a carefully planned comprehensive network management solution that:

- Is scalable and flexible, regardless of how complicated the task is
- Provides around-the-clock availability and reliability of applications to users, no matter where they’re located in the world
- Is capable of building a solid network foundation, despite the complexity of the topology

3.3 Verification guidelines

To ensure that your topology is successful, first verify all individual components and then test the entire topology as a whole. Whenever possible, try to isolate some blocks of your architecture and test them independently.

To conduct your topology test, you need to simulate potential Web users. Some applications, such as IBM Rational® Performance Tester, can help you to do this. You can learn more about Rational Performance Tester on the Web at:

http://www-306.ibm.com/software/awdtools/tester/performance/

Another application to consider for this simulation is OpenSTA. You can download OpenSTA, as an open source project, from the Web at:

http://www.opensta.org/

Part 2

Implementation steps

This part provides the step-by-step implementation of several topologies. It provides enough information for you to perform a full setup and configuration of several topologies based on WebSphere Application Server V6.

4

Scenario 1: Configuration for a small company

This chapter presents the components and the setup steps required to implement a configuration for a small company. It covers the different components, including the firewall, the Web server, and the application server, that make up the topology and the specific Transmission Control Protocol/Internet Protocol (TCP/IP) settings that are used within the network. In addition, the chapter outlines step-by-step how to create the WebSphere and HTTP profiles. Finally, it presents verification steps that you can perform to ensure that the topology is working correctly.

You must review this chapter before you decide if the scenario is compatible to the needs of your business. The chapter takes a step-by-step approach to setting up the sample topology. While you can implement or use some of the steps for your own custom topology, you must consider the dependencies that each component has with other components.

> **Note:** The TCP/IP configuration and all system names presented in this chapter are for example purposes only. While performing these instructions on your own, you must substitute the data for your own system.

4.1 Small company scenario

The topology in this chapter is intended for a small company, although it may be suitable for companies of other sizes as well. This scenario is for companies that do not need to implement a high availability solution, but still want added security. Figure 4-1 illustrates the three-tier approach of the topology that is used in this scenario.

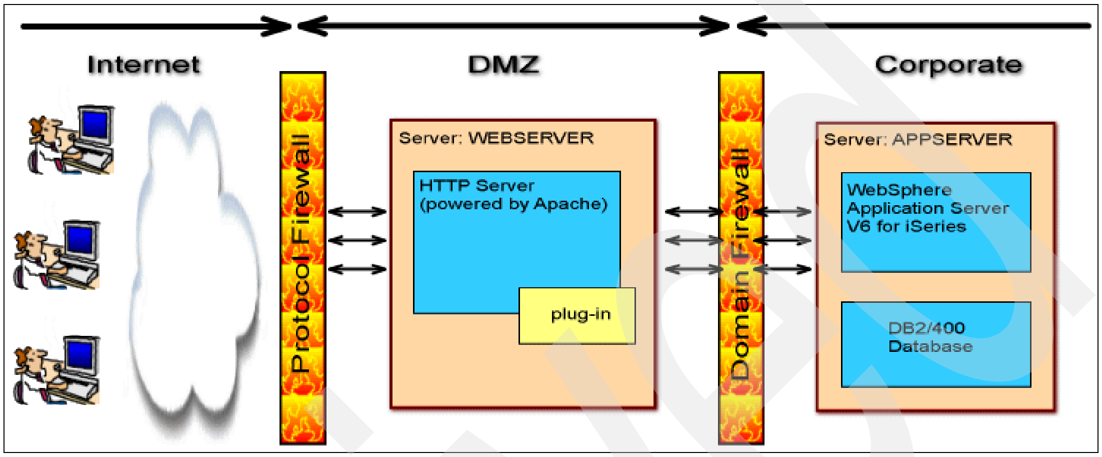


Figure 4-1 Small customer environment

The Web server in Figure 4-1 is referred to as a *remote Web server* because it does not reside on the same system as the WebSphere Application Server. Another important aspect of this remote Web server is the existence of the plug-in component with the configuration file.

The application server is the third tier in this topology. The WebSphere Application Server, as well as the database, reside on this system.

Two firewalls complete the topology. As shown in Figure 4-1, the first firewall acts as a protocol firewall and is positioned between the Internet and the company's demilitarized zone (DMZ). The second firewall, acting as a domain firewall, is positioned between the Web server system and the application server. For more information about firewalls, see 1.1.2, "Firewall" on page 4, as well as 4.3.2, "Firewall configuration" on page 65, for the firewall settings used in this scenario.

Before you decide to implement this type of topology, consider how important a highly available application is to your company. A failure on any of the components in this topology can result in lost business until that time when the failing component is recovered and back in operation. You must do proper performance capacity planning before you implement this scenario. Capacity planning ensures that the company's hardware configuration is adequate to handle the performance characteristics of a particular WebSphere application.

You must also consider that no data redundancy is configured for this topology. Performing system saves on a regular basis is important in this type of configuration for a small company. To maintain database integrity and consistency, in the event of an outage, use commitment control within applications. To assist in the event of an unplanned system outage, you can implement system-managed access path protection (SMAPP). To learn more about network redundancy, see 1.3.7, "Topology with redundancy of several components" on page 18.

If you still have concerns in regard to the type of topology to implement, review "Scenario 2: Configuration for a large company" on page 93, which presents a scenario of a larger and more complex topology. If the topology, as shown in Figure 4-1, is suitable to the needs of the company, then proceed with the sections that follow.

4.2 Installed software

For the small company scenario, WebSphere Application Server V6 software is installed on each system. Table 4-1 lists the edition and features that are installed on each iSeries.

Table 4-1 Installed software for small company scenario

iSeries name	WebSphere Application Server V6 edition	Installed features
WEBSERVER	Express	Web server plug-in only
APPSERVER	Express	All features

To install the software, refer to Appendix B, “Installing WebSphere Application Server Version 6” on page 255. Be sure to install the WebSphere Application Server V6 software on each system before your continue with the next section.

4.3 Topology setup

In this section, you learn how to install and configure an environment, which is suitable for small enterprises or for clients with cost considerations. The topology takes advantage of software that is delivered already with all new eServer i5 systems, such as HTTP Server (powered by Apache) and WebSphere Application Server V6.0 Express for iSeries.

In this scenario, we implement a high level of security; HTTP Server is separated from the WebSphere Application Server and HTTP Server is separated from the Internet. The sample architecture uses two iSeries servers and two firewalls. As an alternative, one iSeries system, with two logical partitions (LPARs) can also be used in this scenario. In either case, the first server works as a Web server, and the second one works as an application server. From the perspective of the application server, the HTTP server is remote.

We explain how to use all the components for this scenario, starting with configuring the network environment, continuing with the firewalls, Web server, and application server, and finishing with the verification steps.

4.3.1 Network configuration

The network infrastructure for computer systems is like the nervous system in the human body. It allows all the components to communicate with each other. You must consider the network infrastructure from the beginning because it influences steps that are required later in the configuration. This is important from both the communication point of view and even more so from the security perspective. By isolating network devices in different subnetworks, you can achieve an appropriate level of security.

Figure 4-2 illustrates the network topology that is used in this scenario for a small customer.

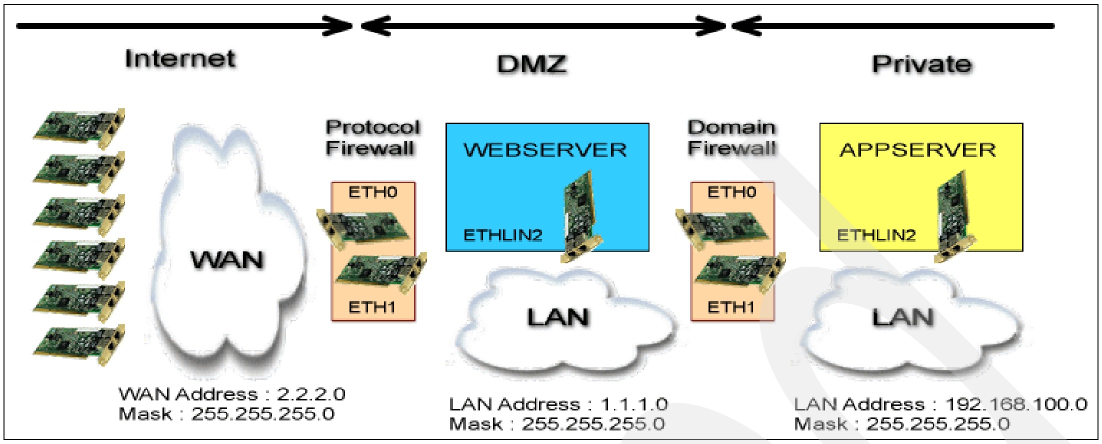


Figure 4-2 Network topology

As shown in Figure 4-2, there are three networks in this scenario. Each network is separated from the next network, through a firewall, and uses a different TCP/IP address range. The topology is made up of the following networks:

- **Private LAN**: This network represents the internal company LAN and is the most secured and protected network. The most important data is processed and stored in this network. No one from an outside network can reach this server directly. This private network contains the application server and database server on the same physical box.
- **Demilitarized zone**: The DMZ represents the network between the two firewalls. Usually, all Web servers, which serve static and less confidential data, are placed in this network. The DMZ has its own IP address range, different from both the private and Internet networks. In this scenario, one HTTP Server is configured in the DMZ.
- **Internet**: This network represents all potential Web users. It is the least secure network because an attack from a potential hacker comes from this network. This network has a different address range from the other two networks. The protocol firewall is the first line of defense for the company intranet.

Table 4-2 shows the network settings for this scenario, including all TCP/IP address ranges and network resources.

Table 4-2 Network inventory

Network	Network address	Network mask	IP address range	Network adapter/device
Internet	2.2.2.0	255.255.255.0	2.2.2.1 - 2.2.2.254	* All web users * ETH0/protocol firewall[a]
DMZ	1.1.1.0	255.255.255.0	1.1.1.1 - 1.1.1.254	* ETH1/protocol firewall[a] * ETH0/ domain firewall[a] * ETHLIN2/WEBSERVER
Private	192.168.100.0	255.255.255.0	192.168.100.1 - 192.168.100.254	* ETH1/domain firewall[a] * ETHLIN2/APPSERVER

[a] The firewalls used in the scenario each have two adapters named ETH0 and ETH1. Ensure that, when either adapter is referred to, the specific firewall is also identified. Refer to Figure 4-2 for the architecture of the network used in this scenario.

Table 4-3 shows the actual IP addresses that are used on each component in the scenario.

Table 4-3 IP address assignment

Network adapter/device	IP address	Port	Mask
ETH0/protocol firewall[a]	2.2.2.1 (Internet side)	80	255.255.255.0
ETH1/protocol firewall[a]	1.1.1.1 (DMZ side)	80	255.255.255.0
ETHLIN2/WEBSERVER	1.1.1.100	80	255.255.255.0
ETH0/domain firewall[a]	1.1.1.253 (DMZ side)	20000	255.255.255.0
ETH1/domain firewall[a]	192.168.100.1 (Private network side)	20000	255.255.255.0
ETHLIN2/APPSERVER	192.168.100.101	20000	255.255.255.0

[a] The firewalls used in the scenario each have two adapters named ETH0 and ETH1. Ensure that, when either adapter is referred to, the specific firewall is also identified. Refer to Figure 4-2 for the architecture of the network used in this scenario.

Table 4-3 shows that two TCP/IP ports are used for each firewall. All the inbound requests, which come form the Internet, reach port 80 of the Web server. To fulfill these client requests, the Web server directs them to the application server. The application server's Web container listens on port 20000.

To allow IP packets to flow between these three networks, you must set up routing information. Routing tables are used by TCP/IP service in all cases when packets must be sent to other networks or subnetworks. Table 4-4 shows the routing rules applied to the Web server and application server.

Table 4-4 Routing rules

Network adapter	Server name	Route destination	Mask	Next hop
ETHLIN2	WEBSERVER	2.2.2.0 (Internet)	255.255.255.0	1.1.1.1 (protocol firewall)
ETHLIN2	WEBSERVER	192.168.100.0 (private LAN)	255.255.255.0	1.1.1.253 (domain firewall)
ETHLIN2	APPSERVER	1.1.1.0 (DMZ)	255.255.255.0	192.168.100.1 (domain firewall)

The first two rules apply to the Web server, because the Web server communicates with two different networks. The first rule represents all the traffic that needs to be sent to the Internet network. The traffic goes through the ETH1 adapter of the protocol firewall, which listens on port 80 with an IP address of 1.1.1.1. The second rule allows packets to reach the private LAN that has a network address of 192.168.100.0. The third rule applies to the application server, which allows packets to leave the private network and reach the DMZ network.

4.3.2 Firewall configuration

A *firewall* is a blockade between a secure internal network and an untrusted network such as the Internet. Most companies use a firewall to connect an internal network safely to the Internet, although a firewall can also be used to secure an internal network from another internal network.

A firewall allows the control of traffic into and out of your network and minimizes the risk of attack to your network. A firewall securely filters all traffic that enters your network so that only specific types of traffic, for specific destinations, can enter. It minimizes the risk that someone could use TELNET or File Transfer Protocol (FTP) to gain access to your internal systems.

This scenario uses two Cisco PIX 506E firewalls. (Although not used in this scenario, the Cisco PIX 501 firewall is also suitable for this scenario). The first firewall controls traffic from the Internet to the DMZ, where the Web server is located. The second firewall is positioned between the DMZ and the private network. This is a secure topology, because even if the first firewall is breached, the second firewall still protects the confidential data in the private network.

You can find technical documentation about Cisco security appliances and their software at the following Web sites:

http://www.cisco.com/en/US/products/hw/vpndevc/ps2030/ps4336/index.html
http://www.cisco.com/en/US/products/hw/vpndevc/ps2030/tsd_products_support_series_home.html
http://www.cisco.com/en/US/products/sw/secursw/ps2120/tsd_products_support_series_home.html

Each of the two firewalls has two Ethernet adapters. Example 4-1 shows the IP address settings for the protocol firewall.

Example 4-1 Protocol firewall IP address settings

```
protocol# show ip addr
System IP Addresses:
        ip address outside 2.2.2.1 255.255.255.0
        ip address inside 1.1.1.1 255.255.255.0
```

According to Table 4-3, the IP address 2.2.2.1 is assigned to the ETH0 adapter of the protocol firewall. This adapter, from the firewall's perspective, is the outside adapter. In Example 4-1, IP address 2.2.2.1 is assigned to the outside adapter. Similarly, the ETH1 adapter of the protocol firewall is seen as the inside adapter, from the firewall's perspective.

Example 4-2 shows the two access list rules defined on the protocol firewall. The first rule permits all Internet hosts to establish connections to the 1.1.1.100 IP address on the well-known HTTP port (port 80). In this scenario, IP address 1.1.1.100 represents the Web server. This rule only allows Internet clients to reach the Web server from their browsers based on HTTP protocol. The second rule denies access to any other IP address and port number in our DMZ.

Example 4-2 Protocol firewall access list rules

```
protocol# show access-list
access-list acl_out; 2 elements
access-list acl_out line 1 permit tcp any host 1.1.1.100 eq www (hitcnt=0)
access-list acl_out line 2 deny ip any any (hitcnt=0)
```

When defining the access list rules for a firewall, it is crucial to consider the order of the rules. In Example 4-2, the *permit* rule is placed before the *deny* rule. If the deny rule had been placed first, the permit rule would never be invoked.

Example 4-3 shows the settings for the domain firewall.

Example 4-3 Domain firewall IP address settings

```
domain# show ip addr
System IP Addresses:
        ip address outside 1.1.1.253 255.255.255.0
        ip address inside 192.168.100.1 255.255.255.0
```

In the same way as described for the protocol firewall, the domain firewall has its ETH0 adapter as the outside adapter and ETH1 as the inside adapter. Refer to Table 4-3 on page 65 again to relate the IP address shown in Example 4-3.

Example 4-4 shows the two access list rules that are defined on the domain firewall. The first rule permits connections from host 1.1.1.100, which is the Web server, to establish a connection to the 192.168.100.101 IP address, which is the application server. Moreover, this rule permits connections on port 20000 only. Port 20000 is the application server's Web container port. The Web server, which is located remotely in this scenario, needs to communicate with the application server which will process dynamic requests. In this scenario, we need to only allow for communication based on the 20000 port. The second rule denies any other connections.

Example 4-4 Domain firewall access list rules

```
domain# show access-list
access-list acl_out; 2 elements
access-list acl_out line 1 permit tcp host 1.1.1.100 host 192.168.100.101 eq 20000
hitcnt=0)
access-list acl_out line 2 deny ip any any (hitcnt=0)
```

When you define the access list rules for a firewall, it is crucial to consider the order of the rules. In Example 4-4, the *permit* rule is placed before the *deny* rule. If the deny rule had been placed first, the permit rule would never be invoked.

Example 4-5 shows a permitted connection log from the protocol firewall. This log shows that the firewall sent through packets from the 2.2.2.50 host to the 1.1.1.100 host based on port 80. This is an example of a regular request to the Web server.

Example 4-5 Permitted connection log example

```
302013: Built inbound TCP connection 2 for outside:2.2.2.50/3676 (2.2.2.50/3676) to
inside:1.1.1.100/80 (1.1.1.100/80)
```

Example 4-6 shows a denied connection log from the protocol firewall. It shows that an outside user, in a potential hacker attack, tried to establish a connection to the Web server based on port 2001, to get access to the administration application. Port 2001 is the default port for IBM Web Administration for iSeries.

Example 4-6 Denied connection log example

```
106023: Deny tcp src outside:2.2.2.50/3677 dst inside:1.1.1.100/2001 by access-group
"acl_out"
```

From the domain firewall, Example 4-7 shows a permitted connection log from the Web server to the application server. The log shows that the domain firewall sent through packets from the 1.1.1.100 host to the 192.168.100.101 host based on port 20000. This is an example of regular communications from the Web server to the application server.

Example 4-7 Permitted connection log example

```
302013: Built inbound TCP connection 0 for outside:1.1.1.100/3492 (1.1.1.1000/3492)
 to inside:192.168.100.101/20000 (192.168.100.101/20000)
```

From the domain firewall, Example 4-8 shows a denied connection log. The log shows a host that is connected to the DMZ, with an IP address of 1.1.1.50, tried to reach the application server based on the 20001 port. Because the WebSphere Application Server profile listens on 20001 port (Admin Console port), we cannot allow anyone, outside the private network, to access the WebSphere administration application. In this case, the domain firewall rejected the request.

Example 4-8 Denied connection log example

```
106023: Deny tcp src outside:1.1.1.50/3493 dst inside:192.168.100.101/20001 by a
ccess-group "acl_out"
```

In addition to these basic rules for the firewall, you can set up more sophisticated rules and monitoring levels. Refer to the technical configuration on the Cisco Web site at:

http://www.cisco.com/en/US/products/hw/vpndevc/ps2030/tsd_products_support_series_home.html

4.3.3 HTTP Server (powered by Apache) installation

The HTTP Server (powered by Apache) product (5722-DG1), as shown in Figure 4-1 on page 62, is installed in the DMZ. In most cases, this product is pre-installed on @server i5 servers. Although the installation process for this product is not described in this redbook, you can find additional information about the HTTP Server in *IBM HTTP Server (powered by Apache): An Integrated Solution for IBM @server iSeries Servers,* SG24-6716.

4.3.4 HTTP Server configuration

Configuring the HTTP Server on iSeries includes several steps. You begin by installing the Web server plug-in component on the system where you configure the HTTP server.

Web server plug-ins installation

In this scenario, we take advantage of the integrated HTTP Server (powered by Apache). To establish a connection between the Web server and application server, you need to install additional software, called *Web server plug-ins*. The Web server plug-ins is a feature of WebSphere Application Server V6. The Web server plug-ins installation is described in Appendix B, “Installing WebSphere Application Server Version 6” on page 255.

After you install the Web server plug-ins feature, you’ll see that the following Integrated File System (IFS) directories exist on the WEBSERVER host:

- /QIBM/ProdData/WebSphere/AppServer/V6/Base
- /QIBM/UserData/WebSphere/AppServer/V6/Base
- /QIBM/UserData/WebSphere/AppServer/V6/service

On the iSeries, the QWAS6 library is created. It is the default location for the Web server plug-ins program.

Starting IBM Web Administration for iSeries

IBM Web Administration for iSeries is a Web-based application that allows you to work with both HTTP server instances and IBM WebSphere Application server instances. Although many more features are available in the IBM Web Administration for iSeries application, we concentrate only on working with these instances.

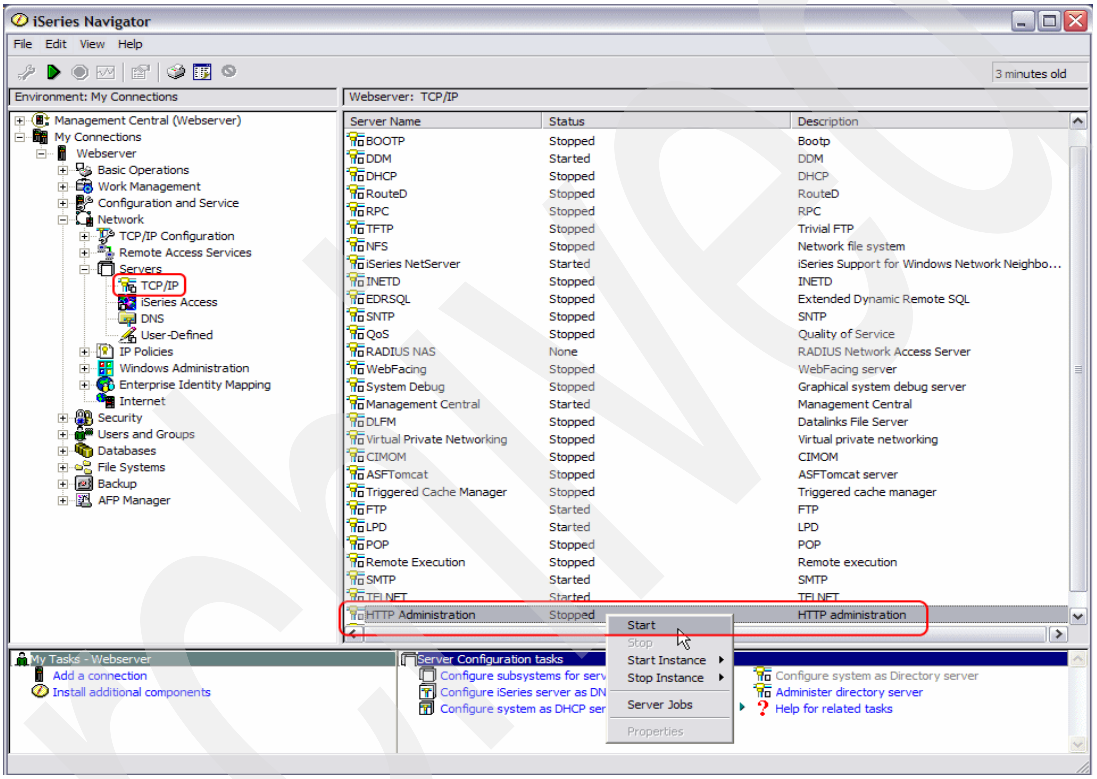
The IBM Web Administration is used to create, administer, and configure an HTTP Server instance.

You can start IBM Web Administration by using one of the two methods. One method is to use OS/400, in which you type the following command:

```
STRTCPSVR SERVER(*HTTP) HTTPSVR(*ADMIN)
```

The other method is to use iSeries Navigator as explained in the following steps. Figure 4-3 shows how to start IBM Web Administration for iSeries from iSeries Navigator.

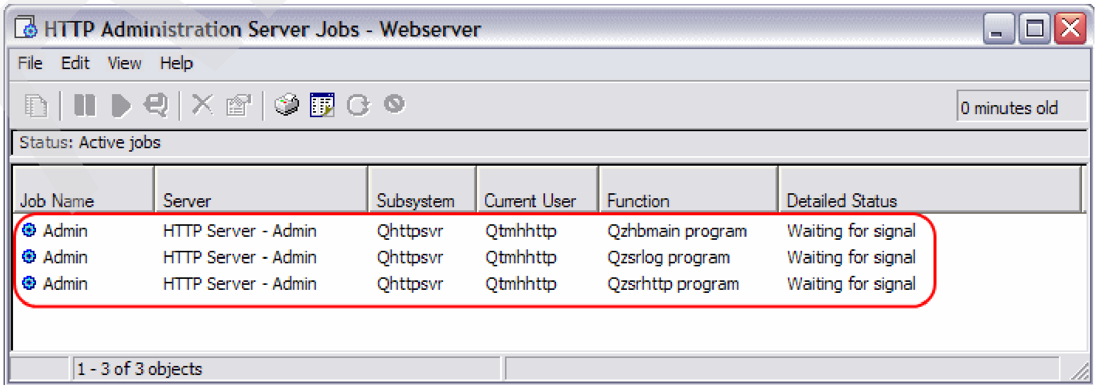
1. In the left panel in iSeries Navigator, expand **My Connections** → ***your iSeries host*** → **Network** → **Servers** and select **TCP/IP**.
2. From the list of servers that is displayed in the right panel, find **HTTP Administration**, right-click, and select **Start**.



Server Name	Status	Description
BOOTP	Stopped	Bootp
DDM	Started	DDM
DHCP	Stopped	DHCP
RouteD	Stopped	RouteD
RPC	Stopped	RPC
TFTP	Stopped	Trivial FTP
NFS	Stopped	Network file system
iSeries NetServer	Started	iSeries Support for Windows Network Neighbo...
INETD	Stopped	INETD
EDRSQL	Stopped	Extended Dynamic Remote SQL
SNTP	Stopped	SNTP
QoS	Stopped	Quality of Service
RADIUS NAS	None	RADIUS Network Access Server
WebFacing	Stopped	WebFacing server
System Debug	Stopped	Graphical system debug server
Management Central	Started	Management Central
DLFM	Stopped	Datalinks File Server
Virtual Private Networking	Stopped	Virtual private networking
CIMOM	Stopped	CIMOM
ASFTomcat	Stopped	ASFTomcat server
Triggered Cache Manager	Stopped	Triggered cache manager
FTP	Started	FTP
LPD	Started	LPD
POP	Stopped	POP
Remote Execution	Stopped	Remote execution
SMTP	Started	SMTP
TELNET	Started	TELNET
HTTP Administration	Stopped	HTTP administration

Figure 4-3 IBM Web Administration for iSeries starting

After IBM Web Administration is started, the QHTTPSVR subsystem is active. Figure 4-4 shows the jobs that are running in this subsystem.



Job Name	Server	Subsystem	Current User	Function	Detailed Status
Admin	HTTP Server - Admin	Qhttpsvr	Qtmhhttp	Qzhbmain program	Waiting for signal
Admin	HTTP Server - Admin	Qhttpsvr	Qtmhhttp	Qzsrlog program	Waiting for signal
Admin	HTTP Server - Admin	Qhttpsvr	Qtmhhttp	Qzsrhttp program	Waiting for signal

Figure 4-4 IBM Web Administration for iSeries jobs

To open the IBM Web Administration GUI, follow these steps:

1. Navigate to the following Uniform Resource Locator (URL) in a browser window:

   `http://hostname:2001/`

   Here *hostname* is the host name or IP address of the iSeries server where the HTTP server has been started. Remember that it you have a firewall installed and configured as described in the previous section, you can access Web Administration for iSeries only from within DMZ.

2. The login window (Figure 4-5) opens. Type a valid OS/400 user profile and password, and click **OK**.

> **Important:** To be authorized to log into the IBM Web Administration for iSeries application, the OS/400 user profile must have *IOSYSCFG special authority.

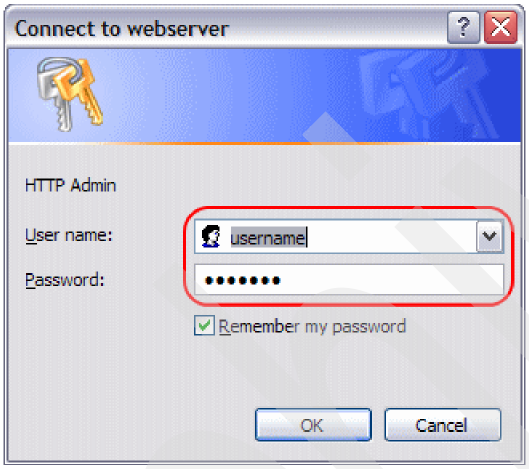
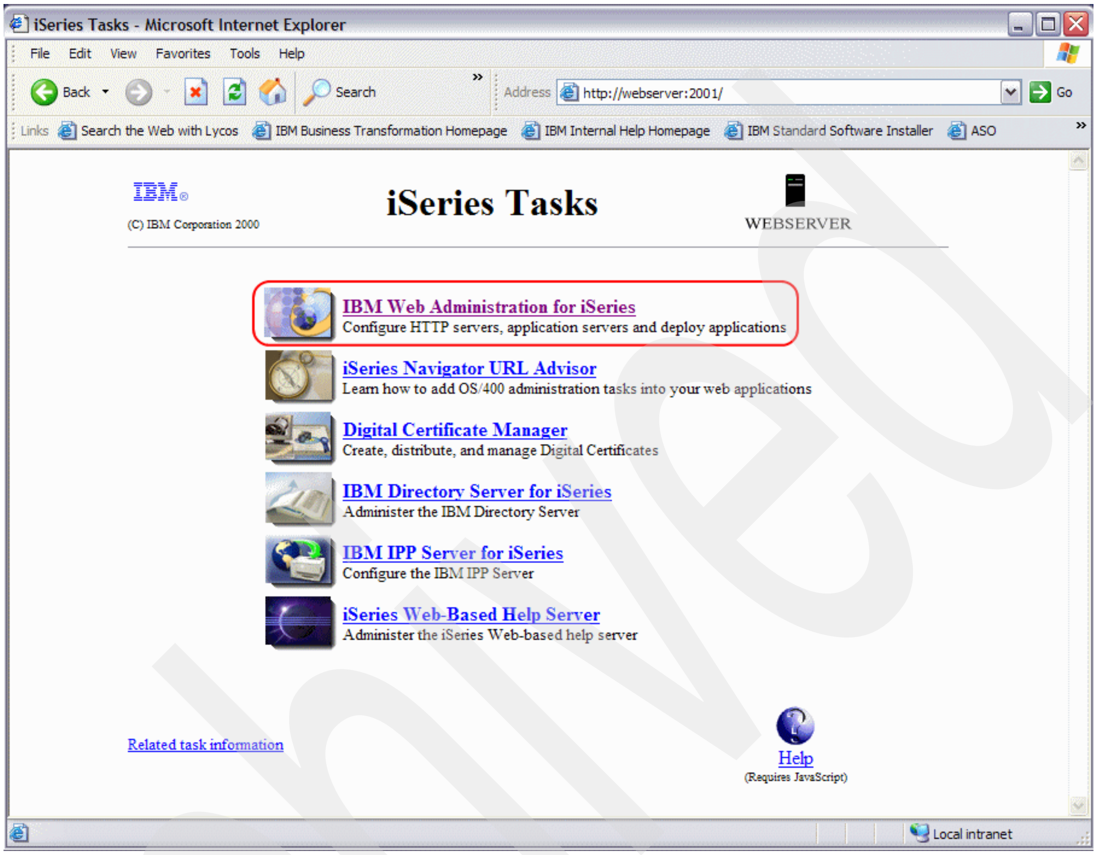


Figure 4-5 IBM Web Administration for iSeries login

3. After a successful login, you see the iSeries Tasks page (Figure 4-6). To work with HTTP Server (powered by Apache) and the IBM WebSphere Application Server instances, click **IBM Web Administration for iSeries**.


iSeries Tasks - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Back
Search
Address
http://webserver:2001/
Go
Links
Search the Web with Lycos
IBM Business Transformation Homepage
IBM Internal Help Homepage
IBM Standard Software Installer
ASO
IBM
WEBSERVER
Local intranet

(C) IBM Corporation 2000

iSeries Tasks

IBM Web Administration for iSeries
Configure HTTP servers, application servers and deploy applications

iSeries Navigator URL Advisor
Learn how to add OS/400 administration tasks into your web applications

Digital Certificate Manager
Create, distribute, and manage Digital Certificates

IBM Directory Server for iSeries
Administer the IBM Directory Server

IBM IPP Server for iSeries
Configure the IBM IPP Server

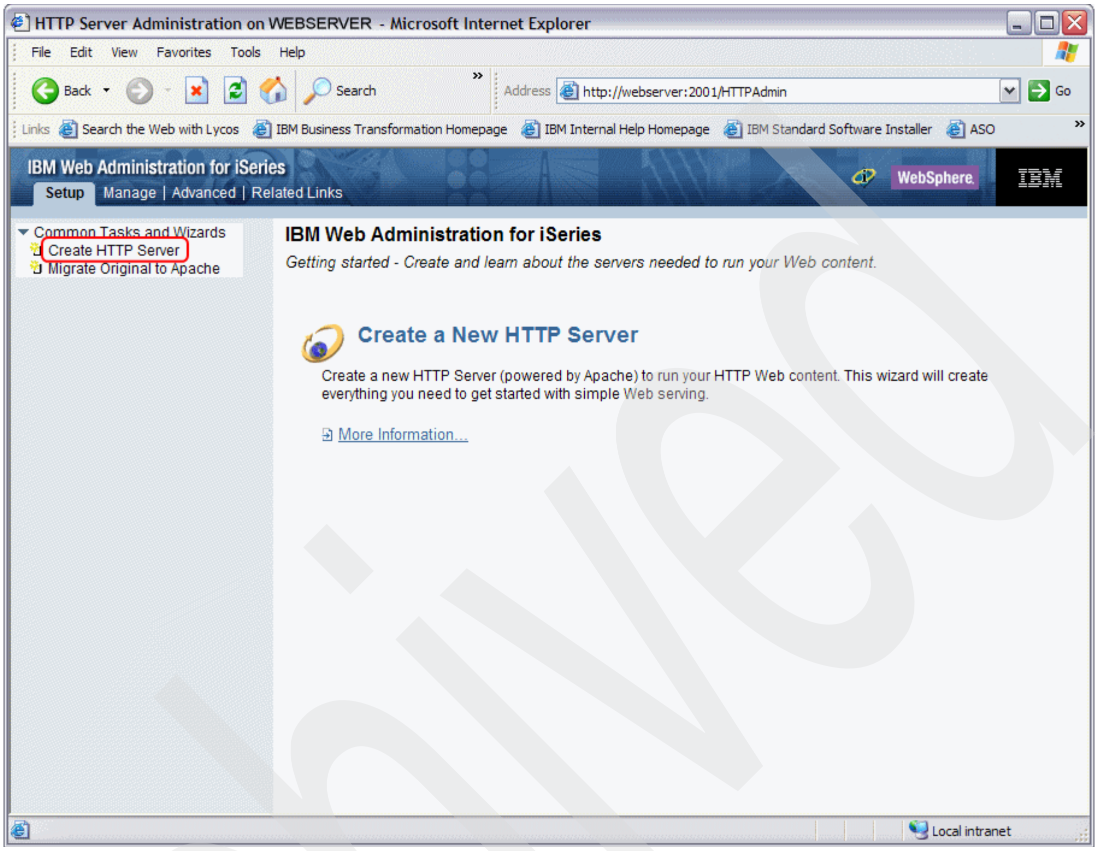
iSeries Web-Based Help Server
Administer the iSeries Web-based help server

Related task information

Help
(Requires JavaScript)

Figure 4-6 IBM Web Administration for iSeries tasks page

The IBM Web Administration for iSeries page (Figure 4-7) opens. If the window is different from the example shown in Figure 4-7, click the **Setup** tab at the top of the page.



HTTP Server Administration on WEBSERVER - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Address http://webserver:2001/HTTPAdmin
Go
Links Search the Web with Lycos IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer ASO
IBM Web Administration for iSeries
Setup | Manage | Advanced | Related Links
WebSphere
IBM
Common Tasks and Wizards
Create HTTP Server
Migrate Original to Apache
Local intranet

IBM Web Administration for iSeries

Getting started - Create and learn about the servers needed to run your Web content.

Create a New HTTP Server

Create a new HTTP Server (powered by Apache) to run your HTTP Web content. This wizard will create everything you need to get started with simple Web serving.

More Information...

Figure 4-7 IBM Web Administration for iSeries

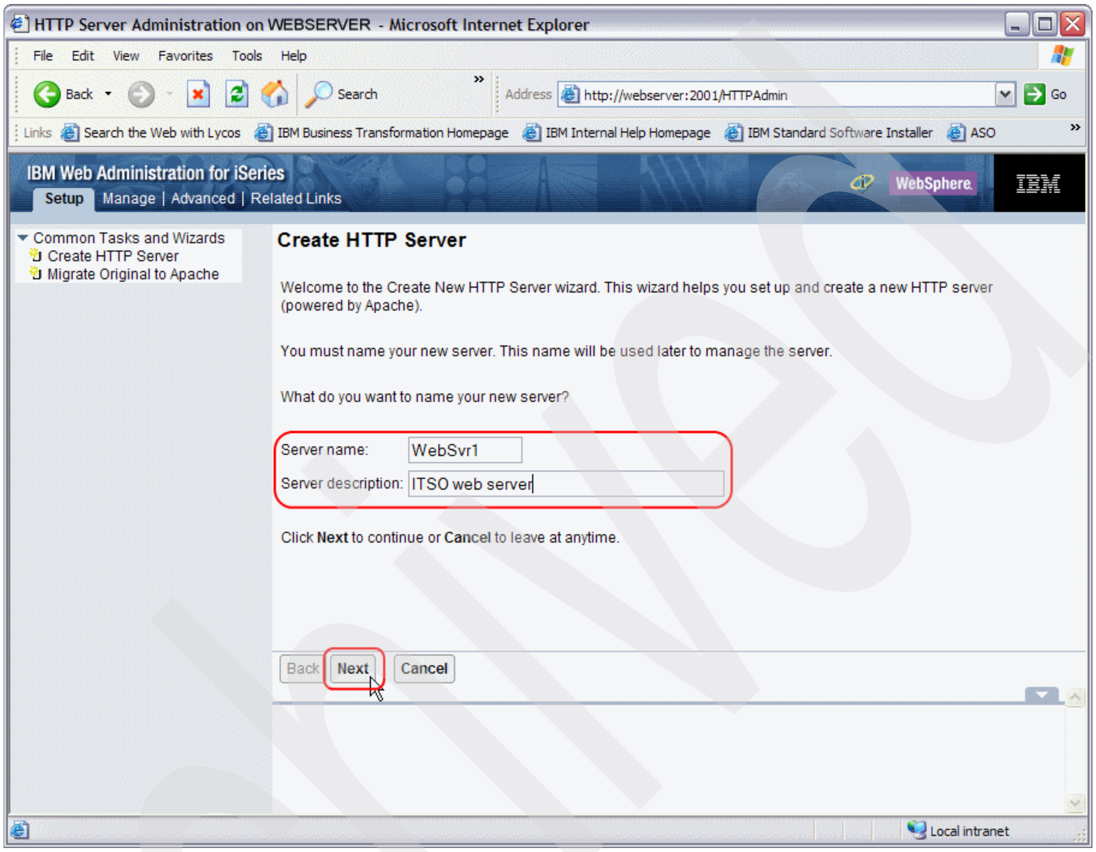
Creating an HTTP Server instance

Next you create a new Web server instance. Table 4-5 contains the parameters that are required for creating an HTTP Server instance.

Table 4-5 HTTP Server instance parameters

Instance name	TCP/IP address	TCP/IP Port	Instance root directory
WebSvr1	1.1.1.100	80	/www/WebSvr1

1. From the IBM Web Administration for iSeries page (Figure 4-7), click **Create HTTP Server**.
2. In the Create HTTP Server panel (Figure 4-8), type the server name. In this case, we type the name from Table 4-5. Optionally, you can enter a server description. Click **Next**.


HTTP Server Administration on WEBSERVER - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Back
Search
Address http://webserver:2001/HTTPAdmin
Go
Links
Search the Web with Lycos
IBM Business Transformation Homepage
IBM Internal Help Homepage
IBM Standard Software Installer
ASO
IBM Web Administration for iSeries
Setup
Manage
Advanced
Related Links
WebSphere
IBM
Common Tasks and Wizards
Create HTTP Server
Migrate Original to Apache
Back
Next
Cancel
Local intranet

Create HTTP Server

Welcome to the Create New HTTP Server wizard. This wizard helps you set up and create a new HTTP server (powered by Apache).

You must name your new server. This name will be used later to manage the server.

What do you want to name your new server?

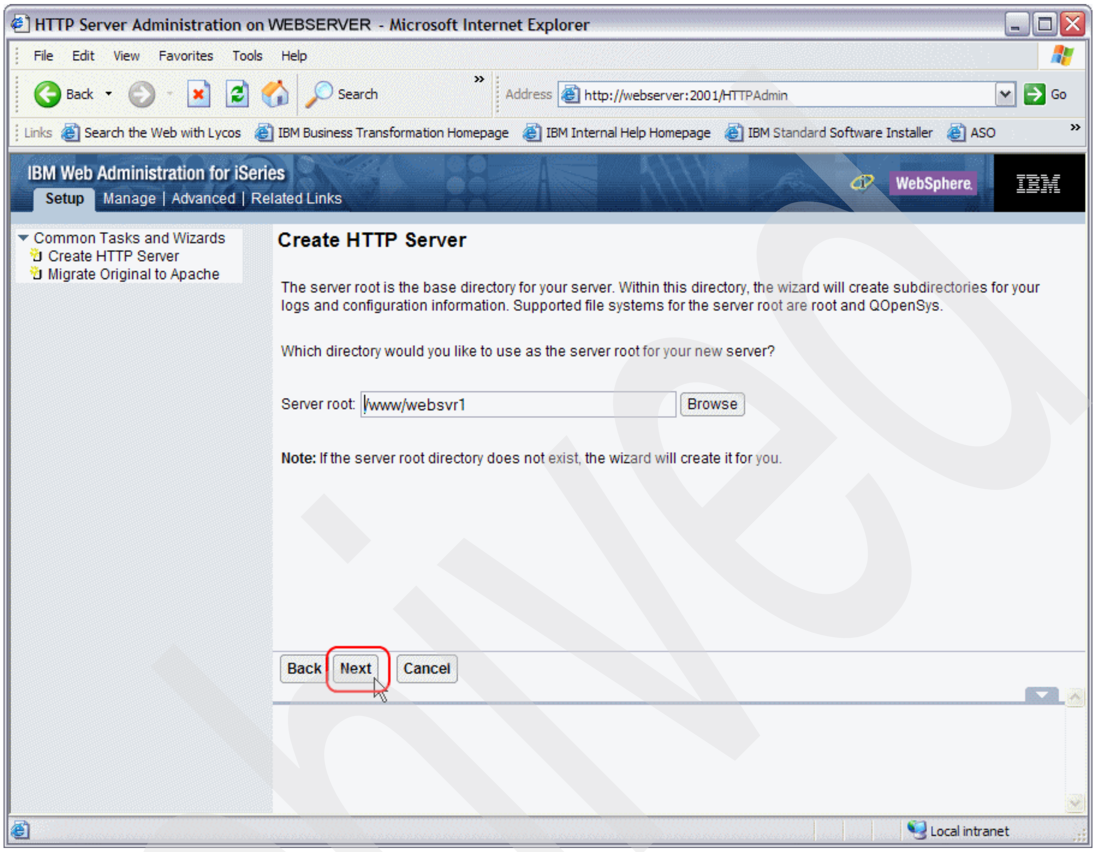
Server name: WebSvr1

Server description: ITSO web server

Click **Next** to continue or **Cancel** to leave at anytime.

Figure 4-8 Create HTTP Server: Naming the server

3. On the next panel (Figure 4-9), specify the directory that you want to use as the server root. You can accept the default HTTP server root directory, or you can enter a new one. In this scenario, we use the default value. Click **Next**.


HTTP Server Administration on WEBSERVER - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Address http://webserver:2001/HTTPAdmin
Go
Links Search the Web with Lycos IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer ASO
IBM Web Administration for iSeries
Setup Manage | Advanced | Related Links
WebSphere
IBM
Common Tasks and Wizards
Create HTTP Server
Migrate Original to Apache
Browse
Back
Next
Cancel
Local intranet

Create HTTP Server

The server root is the base directory for your server. Within this directory, the wizard will create subdirectories for your logs and configuration information. Supported file systems for the server root are root and QOpenSys.

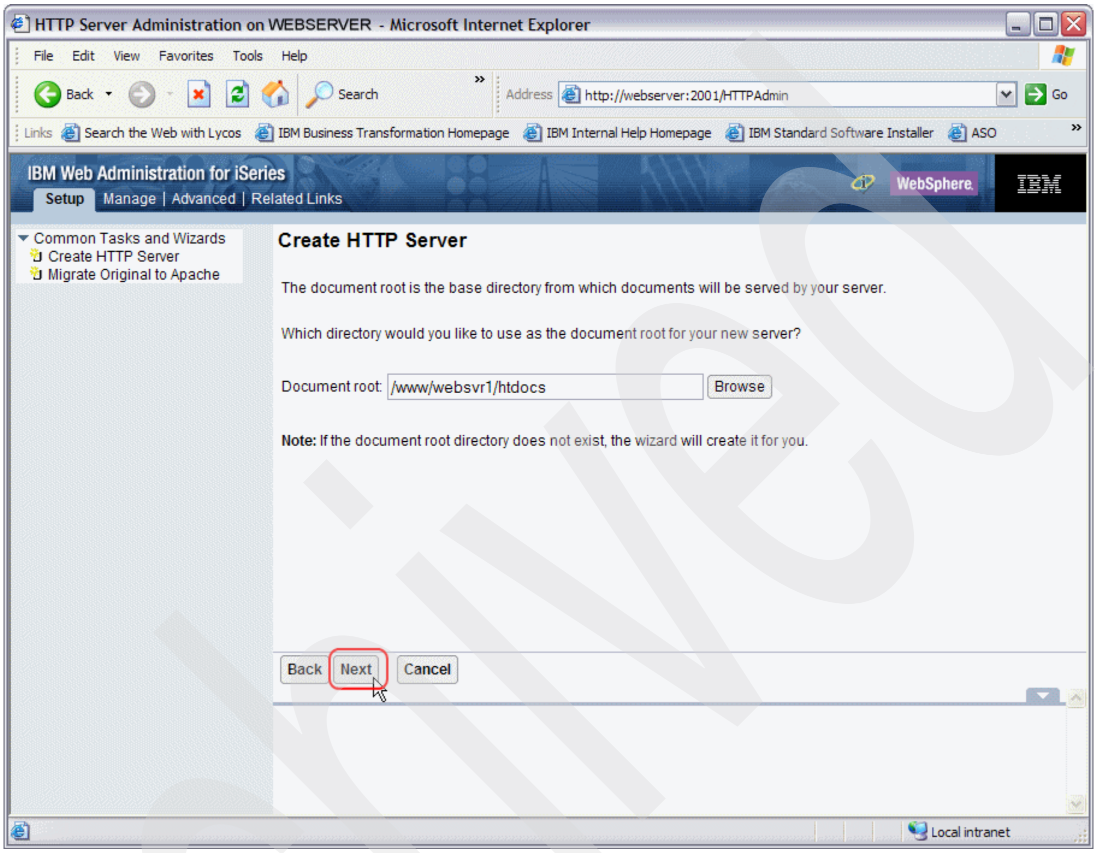
Which directory would you like to use as the server root for your new server?

Server root: /www/websvr1

Note: If the server root directory does not exist, the wizard will create it for you.

Figure 4-9 Create HTTP Server: Specifying the server root directory

4. On the next panel (Figure 4-10), specify the directory that you want to use for the document root. The document root is the root directory for static Web content. You can accept the default document root, or enter a new one. In this scenario, we use the default value. Click **Next**.



HTTP Server Administration on WEBSERVER - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Address http://webserver:2001/HTTPAdmin
Go
Links Search the Web with Lycos IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer ASO
IBM Web Administration for iSeries
Setup | Manage | Advanced | Related Links
WebSphere
IBM
Common Tasks and Wizards
Create HTTP Server
Migrate Original to Apache
Back
Next
Cancel
Local intranet

Create HTTP Server

The document root is the base directory from which documents will be served by your server.

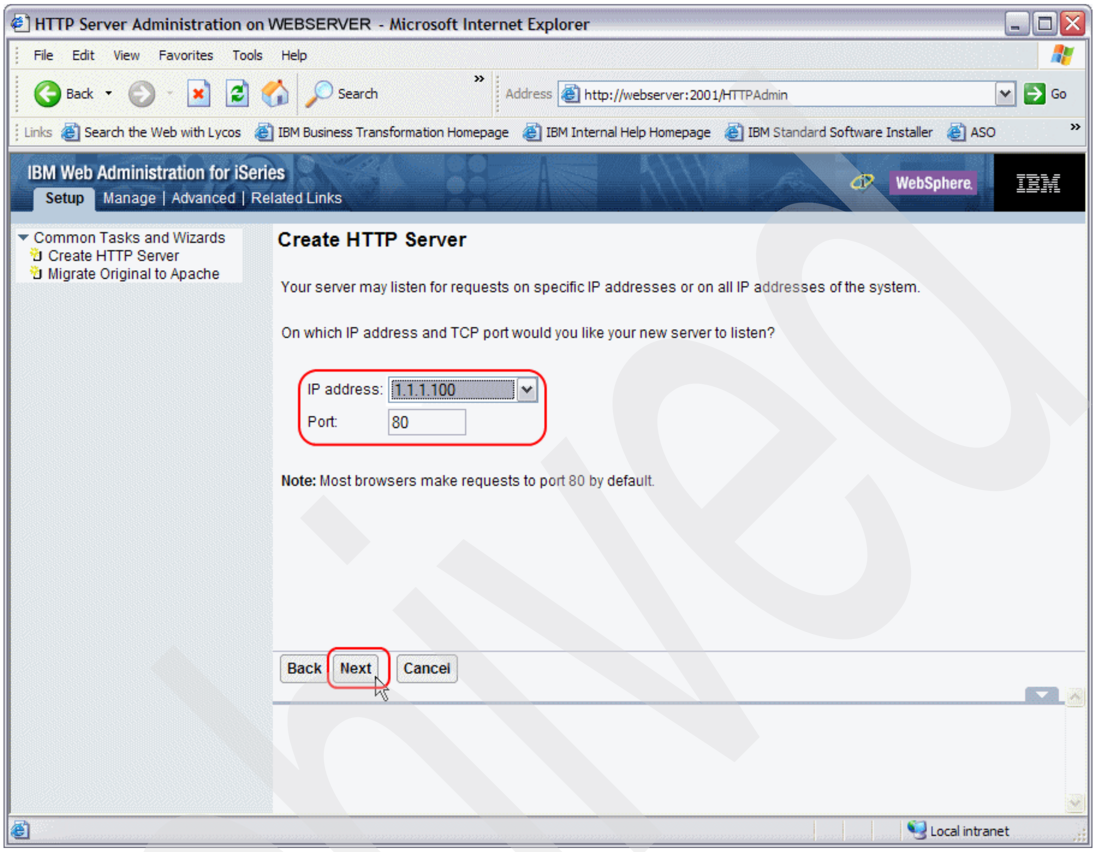
Which directory would you like to use as the document root for your new server?

Document root: /www/websvr1/htdocs [Browse]

Note: If the document root directory does not exist, the wizard will create it for you.

Figure 4-10 Create HTTP Server: Specifying the document root directory

5. On the next panel (Figure 4-11), specify the TCP/IP address and the port on which the server will listen. In this scenario, we enter the values from Table 4-5 on page 72. Click **Next**.


HTTP Server Administration on WEBSERVER - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Address http://webserver:2001/HTTPAdmin
Go
Links Search the Web with Lycos IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer ASO
IBM Web Administration for iSeries
Setup | Manage | Advanced | Related Links
WebSphere
IBM
Common Tasks and Wizards
Create HTTP Server
Migrate Original to Apache
Back
Next
Cancel
Local intranet

Create HTTP Server

Your server may listen for requests on specific IP addresses or on all IP addresses of the system.

On which IP address and TCP port would you like your new server to listen?

IP address: 1.1.1.100

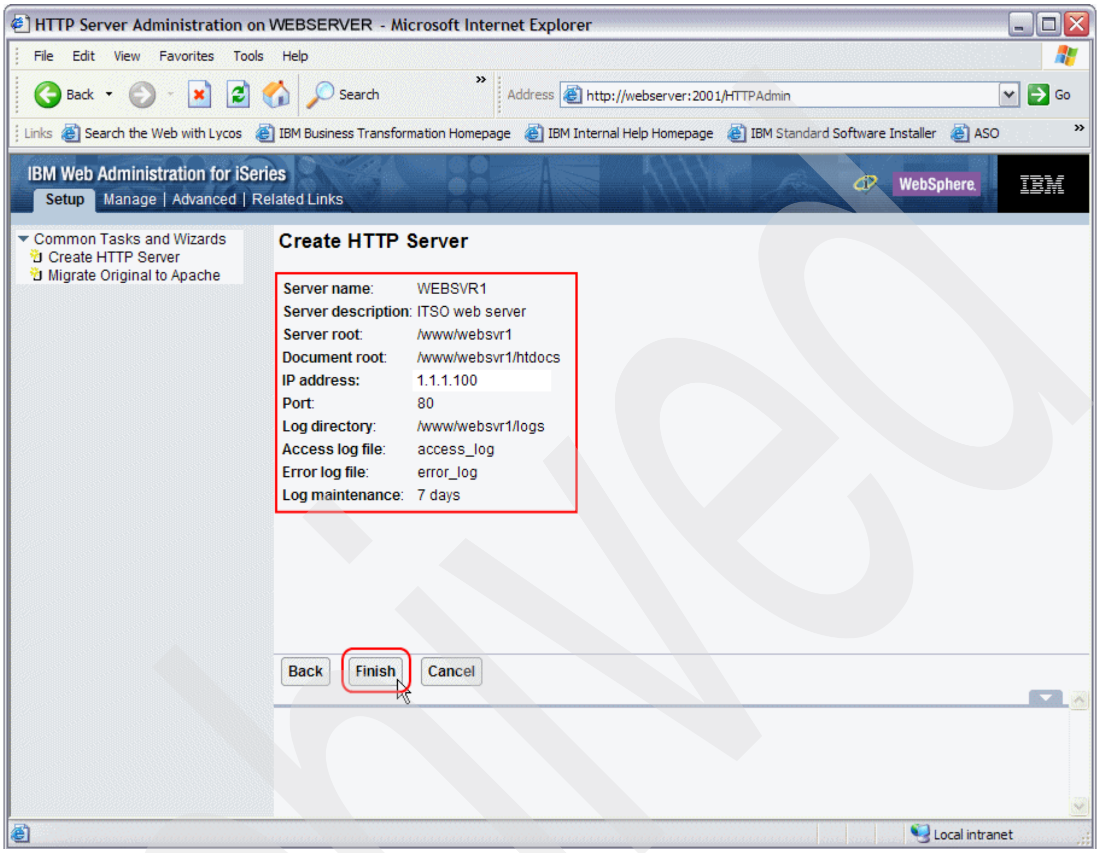
Port: 80

Note: Most browsers make requests to port 80 by default.

Figure 4-11 Create HTTP Server: Specifying the IP address and TCP port

6. The next two panels deal with the Access Log and the time to keep log files. These windows are not shown here. For this scenario, we accepted the default values on these windows. Click **Next** on both panels to continue.

7. The last panel of the HTTP Server creation (Figure 4-12) summarizes the criteria that you defined. Review the information as shown in the window, and verify that it is correct. Then click **Finish**, and a new HTTP server instance is created.



HTTP Server Administration on WEBSERVER - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Address http://webserver:2001/HTTPAdmin
Go
Links Search the Web with Lycos IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer ASO
IBM Web Administration for iSeries
Setup Manage | Advanced | Related Links
WebSphere
IBM
Common Tasks and Wizards
Create HTTP Server
Migrate Original to Apache
Back
Finish
Cancel
Local intranet

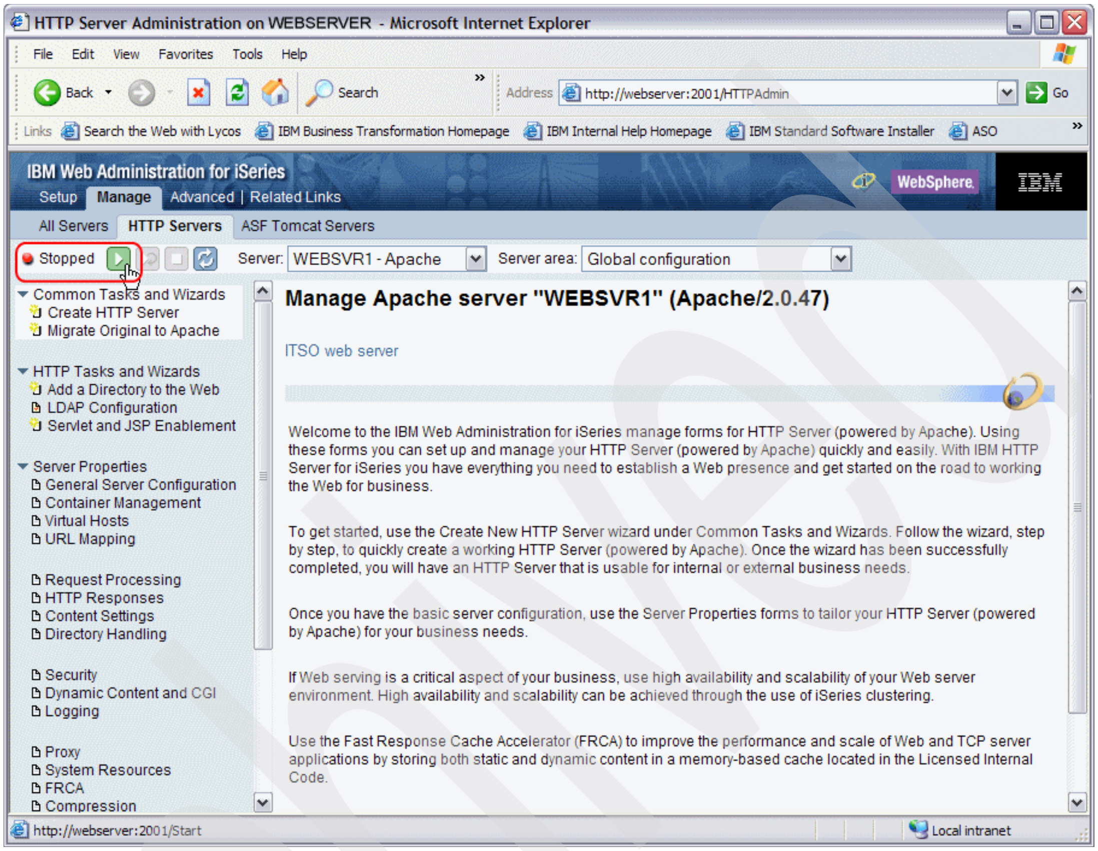
Create HTTP Server

Server name:	WEBSVR1
Server description:	ITSO web server
Server root:	/www/websvr1
Document root:	/www/websvr1/htdocs
IP address:	1.1.1.100
Port:	80
Log directory:	/www/websvr1/logs
Access log file:	access_log
Error log file:	error_log
Log maintenance:	7 days

Figure 4-12 Create HTTP Server: Summary window

Starting and testing the HTTP Server instance

After the HTTP Server instance is created, you see a page similar to the example shown in Figure 4-13. After an instance is created, the default status is *Stopped*.



HTTP Server Administration on WEBSERVER - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Address http://webserver:2001/HTTPAdmin
Go
Links Search the Web with Lycos IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer ASO
IBM Web Administration for iSeries
Setup Manage Advanced | Related Links
WebSphere
IBM
All Servers HTTP Servers ASF Tomcat Servers
Stopped
Server: WEBSVR1 - Apache
Server area: Global configuration
Common Tasks and Wizards
Create HTTP Server
Migrate Original to Apache
HTTP Tasks and Wizards
Add a Directory to the Web
LDAP Configuration
Servlet and JSP Enablement
Server Properties
General Server Configuration
Container Management
Virtual Hosts
URL Mapping
Request Processing
HTTP Responses
Content Settings
Directory Handling
Security
Dynamic Content and CGI
Logging
Proxy
System Resources
FRCA
Compression
http://webserver:2001/Start
Local intranet

Manage Apache server "WEBSVR1" (Apache/2.0.47)

ITSO web server

Welcome to the IBM Web Administration for iSeries manage forms for HTTP Server (powered by Apache). Using these forms you can set up and manage your HTTP Server (powered by Apache) quickly and easily. With IBM HTTP Server for iSeries you have everything you need to establish a Web presence and get started on the road to working the Web for business.

To get started, use the Create New HTTP Server wizard under Common Tasks and Wizards. Follow the wizard, step by step, to quickly create a working HTTP Server (powered by Apache). Once the wizard has been successfully completed, you will have an HTTP Server that is usable for internal or external business needs.

Once you have the basic server configuration, use the Server Properties forms to tailor your HTTP Server (powered by Apache) for your business needs.

If Web serving is a critical aspect of your business, use high availability and scalability of your Web server environment. High availability and scalability can be achieved through the use of iSeries clustering.

Use the Fast Response Cache Accelerator (FRCA) to improve the performance and scale of Web and TCP server applications by storing both static and dynamic content in a memory-based cache located in the Licensed Internal Code.

Figure 4-13 Starting the HTTP Server instance

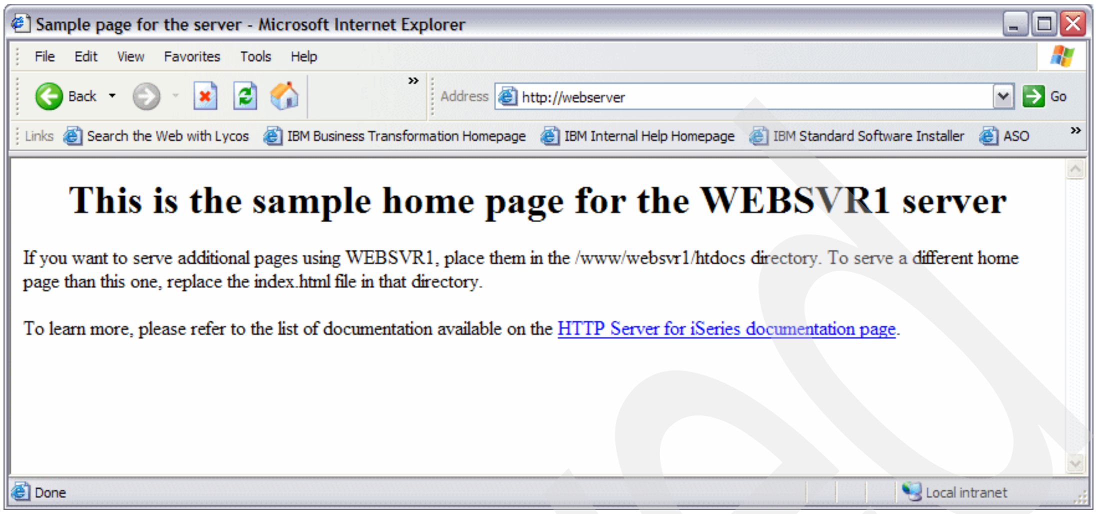
Click the green button as shown in Figure 4-13 to start the instance. When the instance is started, its status changes to *Running*.

To verify that the instance works, open a browser and enter the following URL:

```
http://web_server_host_ name:port
```

Here, *web_server_host_name* is the host name or TCP/IP address of the Web server. *port* is the port on which the HTTP server listens for the request. If the port is 80, then you can omit it in the URL.

If you see the sample home page for the Web server instance, as shown in Figure 4-14, the Web server instance is working properly.


Sample page for the server - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Back
Address
http://webserver
Go
Links
Search the Web with Lycos
IBM Business Transformation Homepage
IBM Internal Help Homepage
IBM Standard Software Installer
ASO
Done
Local intranet

This is the sample home page for the WEBSVR1 server

If you want to serve additional pages using WEBSVR1, place them in the /www/websvr1/htdocs directory. To serve a different home page than this one, replace the index.html file in that directory.

To learn more, please refer to the list of documentation available on the HTTP Server for iSeries documentation page.

Figure 4-14 HTTP server instance home page

4.3.5 WebSphere Application Server profile

> **Attention:** The HTTP server and WebSphere Application server are set up on two separate systems or LPARs. Pay attention to the system on which you should execute each command.

The next step is to create a WebSphere Application Server profile on APPSERVER iSeries. There are several methods to create a WebSphere Application Server profile. You can use either a Qshell command or IBM Web Administration for iSeries. For this scenario, we use Qshell Interpreter to create the WebSphere Application Server profile.

Table 4-6 contains the parameters that we need to create a WebSphere Application Server profile.

Table 4-6 WebSphere Application Server profile parameters

Profile name	TCP/IP address	Starting port
AppSvr1	192.168.100.101	20000

The wasprofile script is used within Qshell to create new server directories and to set up correct authorities. To run this script, the iSeries user profile must have *ALLOBJ authority. Complete the following steps:

1. On the OS/400 command line, enter the Start Qshell (`STRQSH`) command.
2. On the Qshell command line, type the following command:

   ```
   cd /QIBM/ProdData/WebSphere/AppServer/V6/Base/bin
   ```

3. Run the wasprofile script. For this scenario, we enter the following command, in which you replace the parameters in italics with your own values.

   ```
   wasprofile -create -profileName AppSvr1 -templatePath default -startingPort 20000
   ```

> **Note:** The templatePath parameter assumes that the specified template is in the *install_root*/profileTemplates directory. In this scenario, the default profile template is found in the /QIBM/ProdData/WebSphere/AppServer/V6/Base/profileTemplates directory.

After the application server profile is created, you see a message similar to the one in Example 4-9 in the Qshell session.

Example 4-9 wasprofile command output

```
> wasprofile -create -profileName AppSvr1 -templatePath default -startingPort 20000
  INSTCONFSUCCESS: Success: The profile now exists.
  $
```

Starting the WebSphere Application Server profile

To start the new WebSphere Application Server profile, from the OS/400 command line, perform the following step on the APPSERVER iSeries:

1. On the OS/400 command line, type the `STRQSH` command.
2. On the Qshell command line, type the following command:

   ```
   cd /QIBM/ProdData/WebSphere/AppServer/V6/Base/bin
   ```

3. Run the startServer script. For this scenario, the command is:

   ```
   startServer -profileName AppSvr1
   ```

   Here *AppSvr1* is the application server profile created in “WebSphere Application Server profile” on page 79.

After the application server profile is started, you see messages similar to the ones in Example 4-10 in the Qshell session.

Example 4-10 startServer command output

```
> startServer -profileName AppSvr1
  CPC1221: Job 016634/QEJBSVR/APPSVR1 submitted to job queue QWASJOBQ in
  library QWAS6.
  WAS6123: Application server started.
   Cause . . . . . :   Application server AppSvr1 in profile (instance)
     AppSvr1 has started and is ready to accept connections on admin port
     20001.
  $
```

> **Note:** The admin port for the WebSphere Application Server profile is shown in the output for the startServer command. This port is required when you open the WebSphere Administrative Console.

As a result of the startServer command, one job in the QWAS6 subsystem will be started. The name of the job is the name of the application server profile (AppSvr1).

Testing the WebSphere Application Server profile

The installation verification test (IVT) script verifies that the application server for a profile is functioning correctly. For a WebSphere Application Server profile, the IVT script verifies that a JavaServer Page (JSP), a servlet, and an Enterprise JavaBean (EJB) can be successfully invoked.

To run the IVT script, complete the following steps:

1. On the OS/400 command line, enter the STRQSH command.
2. On the Qshell command line, type the following command:

   ```
   cd /QIBM/ProdData/WebSphere/AppServer/V6/Base/bin
   ```

3. Run the IVT script. For this scenario, the command is:

   ```
   ivt AppSvr1 AppSvr1
   ```

   Here the *AppSvr1* parameters represent both the server name and profile name. In this scenario, a server name was not specified during the **`wasprofile -create`** command. By default, the server name has the same name as the profile name.

After the IVT script is invoked, you see messages similar to those shown in Example 4-11 in the Qshell session.

Example 4-11 IVT command output

```
IVTL0050I: Servlet Engine Verification Status - Passed
Testing server using the following
URL:http://APPSERVER:20000/ivt/ivtserver?parm2=ivtAddition.jsp
IVTL0055I: JSP Verification Status - Passed
Testing server using the following URL:http://APPSERVER:20000/ivt/ivtserver?parm2=ivtejb
IVTL0060I: EJB Verification Status - Passed
IVTL0035I: Scanning the file
/QIBM/UserData/WebSphere/AppServer/V6/Base/profiles/AppSvr1/logs/AppSvr1/SystemOut.log for
errors and warnings
IVTL0040I: 0 errors/warnings were detected in the file
/QIBM/UserData/WebSphere/AppServer/V6/Base/profiles/AppSvr1/logs/AppSvr1/SystemOut.log
IVTL0070I: IVT Verification Succeeded
IVTL0080I: Installation Verification is complete
$
```

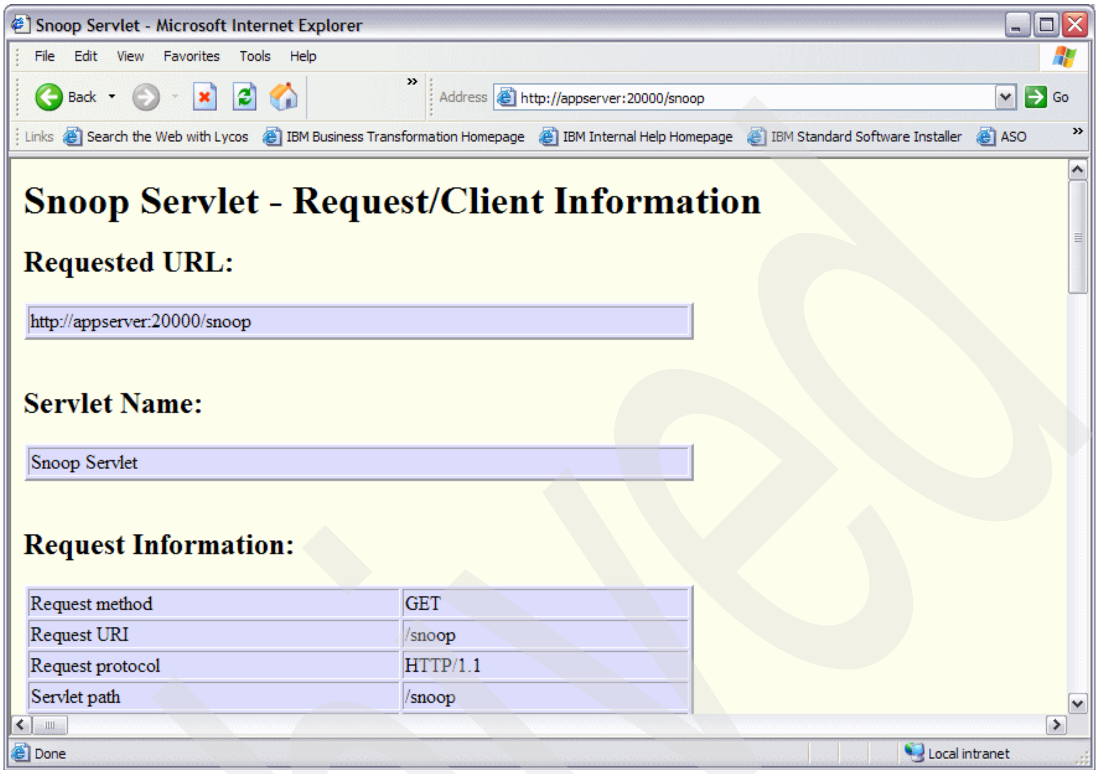
The Web container can be tested through the Snoop servlet. By default, the servlet is mapped to the default virtual host, which listens on the first port in the port range. Looking at Table 4-6 on page 79, the starting port is port 20000, which means that the Snoop servlet will also be available on port 20000.

To test the Snoop servlet, open a browser window and specify the following URL:

```
http://appserver:20000/Snoop
```

Here *appserver* is the name of the iSeries system where the application server is running.

If the Snoop servlet is available and there are no configuration problems, you see a window similar to the one in Figure 4-15.


Snoop Servlet - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://appserver:20000/snoop
Go
Links Search the Web with Lycos IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer ASO
Done
Local intranet

Snoop Servlet - Request/Client Information

Requested URL:

http://appserver:20000/snoop

Servlet Name:

Snoop Servlet

Request Information:

Request method	GET
Request URI	/snoop
Request protocol	HTTP/1.1
Servlet path	/snoop

Figure 4-15 Snoop servlet output

4.3.6 Associating the HTTP server with the WebSphere profile

After testing the HTTP Server instance and the WebSphere Application Server instance, you must associate these two components.

Creating the Web server definition

To establish a connection between the HTTP Server instance and WebSphere Application Server profile, you need to create a Web server definition on the iSeries application server, which in this scenario is the iSeries AppServer server.

Before you create a Web server definition, you need the parameters required to run the configuration script. Table 4-7 list parameters that are used to run the configuration script.

Table 4-7 Web server definition script parameters

WebSphere profile name	Web server instance name	Web server instance port	Web server definition name
AppSvr1	WebSvr1	80	IHS_WEBSERVER_WebSvr1

In Table 4-7, the first parameter (AppSvr1) was selected in 4.3.5, “WebSphere Application Server profile” on page 79. The second and third parameters (WebSvr1 and 80) were selected in “Creating an HTTP Server instance” on page 72. The fourth parameter is the Web server definition name. It has to be created according to the following convention:

webserverType_hostName_webserverInstanceName

Note the following explanation:

- *webserverType*: This refers to the type of Web server. A value of IHS or DOMINO is accepted. For this scenario, the type is IHS for IBM HTTP Server.
- *hostName*: This is the Web server iSeries host name.
- *webserverInstanceName*: This is the HTTP Server instance name.

Based on the parameters in Table 4-7, complete the following steps to create a Web server definition:

1. On the OS/400 command line, enter the `STRQSH` command.
2. On the Qshell command line, type the following command:

   ```
   cd /QIBM/ProdData/WebSphere/AppServer/V6/Base/bin
   ```

3. Run the configureOs400WebServerDefinition script. For this scenario, the command is:

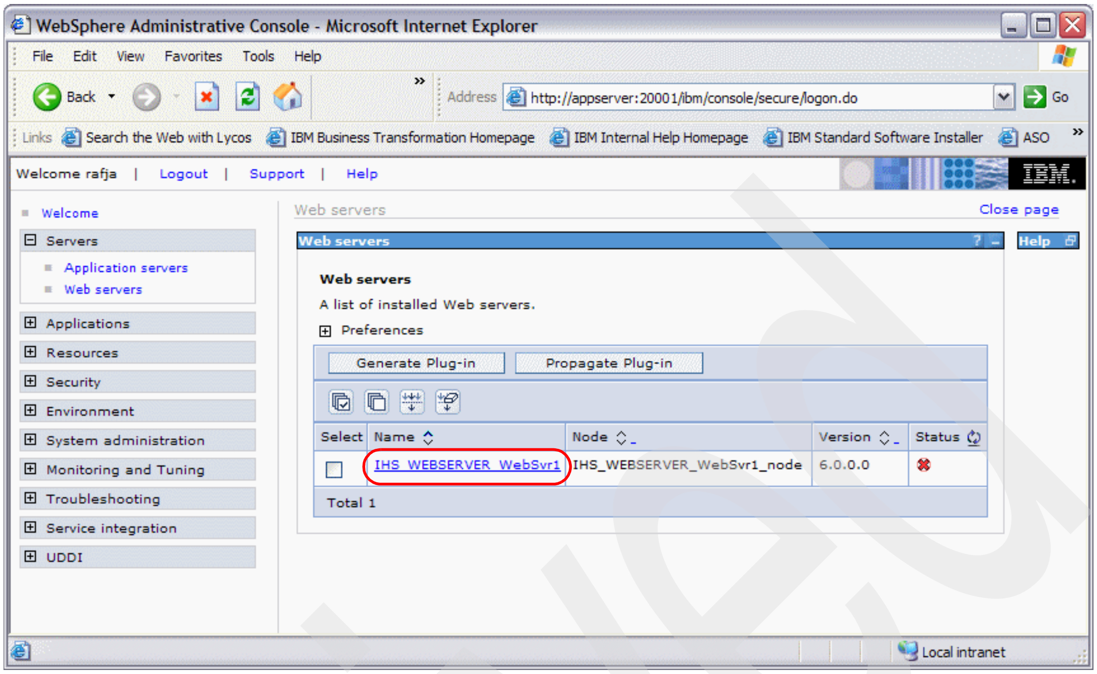
   ```
   configureOs400WebServerDefinition -profileName AppSvr1 -webserver.instance.name WebSvr1
   -webserver.port 80 -webserver.name IHS_WEBSERVER_WebSvr1
   ```

After the script is invoked, you see messages similar to the ones in Example 4-12, in the Qshell session.

Example 4-12 Web server definition configuration output

```
> configureOs400WebServerDefinition -profileName AppSvr1 -webserver.instance.name WebSvr1
-webserver.port 80 -webserver.name IHS_WEBSERVER_WebSvr1
  ...
Input parameters:
    Web server name             - IHS_WEBSERVER_WebSvr1
    Web server type            - IHS
    Web server install location - WebSvr1
    Web server config location  - /www/WebSvr1/conf/httpd.conf
    Web server port            - 80
    Web server admin port      - 2001
    Map Applications           - MAP_ALL
    Plugin install location     -
/QIBM/UserData/WebSphere/AppServer/V6/Base/profiles/AppSvr1
    Web server node type       - unmanaged
    Web server node name       - IHS_WEBSERVER_WebSvr1_node
    Web server host name        - WEBSEVER
    Web server operating system - os400

  Creating the unmanaged node IHS_WEBSERVER_WebSvr1_node .
  Unmanged node IHS_WEBSERVER_WebSvr1_node is created.

  Creating the web server definition for IHS_WEBSERVER_WebSvr1.
Target mapping is updated for the application query.
  Start saving the configuration.
  Configuration save is complete.
  $
```

You can also see the result of running this script in the WebSphere Administrative Console. See Figure 4-16. To access this console, see “Updating the virtual hosts” on page 84.


WebSphere Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://appserver:20001/ibm/console/secure/logon.do
Go
Links Search the Web with Lycos IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer ASO
Welcome rafja | Logout | Support | Help
IBM.
Welcome
Servers
Application servers
Web servers
Applications
Resources
Security
Environment
System administration
Monitoring and Tuning
Troubleshooting
Service integration
UDDI
Web servers
Close page
Web servers
Help
Preferences
Generate Plug-in
Propagate Plug-in
Local intranet

Web servers

A list of installed Web servers.

Select	Name	Node	Version	Status
□	IHS_WEBSERVER_WebSvr1	IHS_WEBSERVER_WebSvr1_node	6.0.0.0	✖
Total 1				

Figure 4-16 Web server definition

Now that the Web server definition is created, within the application server profile, you must update the virtual hosts.

Updating the virtual hosts

The concept of virtual hosts, in terms of Web serving, refers to the practice of maintaining more than one domain in a single server. Domains are differentiated by their host name or IP address. Client requests are routed to the correct domain by IP address or by the host name contained in the URL header. Virtual hosts control which Web servers can access which application servers.

1. Access the WebSphere Administrative Console from a browser by specifying the following URL:

   ```
   http://hostName:port/ibm/console
   ```

   Note the following explanation:

   – *hostName* is the application server host name

   – *port* is the admin port number. This port number is shown when the profile is started. Refer to Example 4-10 on page 80.

   For this scenario, we used the following URL:

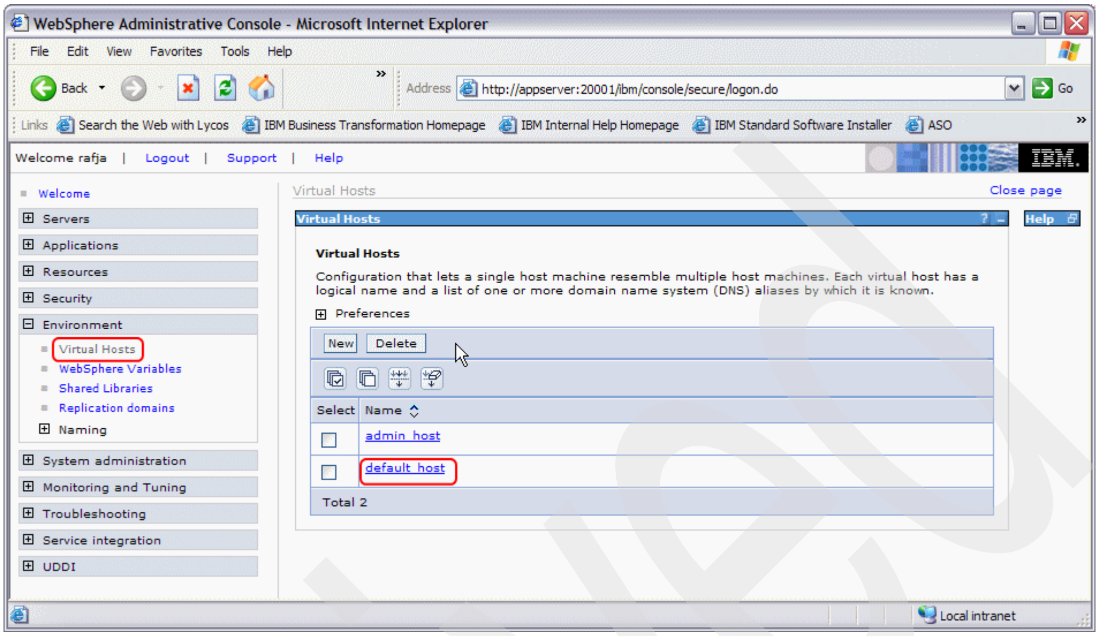
   ```
   http://appserver:20001/ibm/console
   ```

2. By default all WebSphere Application Server profiles come with two virtual hosts already defined as shown in Figure 4-17. The first virtual host is called *admin_host* and is dedicated to the administration console applications. All other applications, by default, use the virtual host called *default_host*. These virtual hosts can be seen in the WebSphere Administrative Console.

3. Sign in to the console.

4. As shown in Figure 4-17, in the left navigation area, expand **Environment** and then click **Virtual Hosts**.

5. In the Virtual Hosts panel on the right (Figure 4-17), click **default_host**.


WebSphere Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://appserver:20001/ibm/console/secure/logon.do
Go
Links Search the Web with Lycos IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer ASO
Welcome rafja | Logout | Support | Help
IBM.
Welcome
Servers
Applications
Resources
Security
Environment
Virtual Hosts
WebSphere Variables
Shared Libraries
Replication domains
Naming
System administration
Monitoring and Tuning
Troubleshooting
Service integration
UDDI
Virtual Hosts
Close page
Help
Preferences
New
Delete
Local intranet

Virtual Hosts

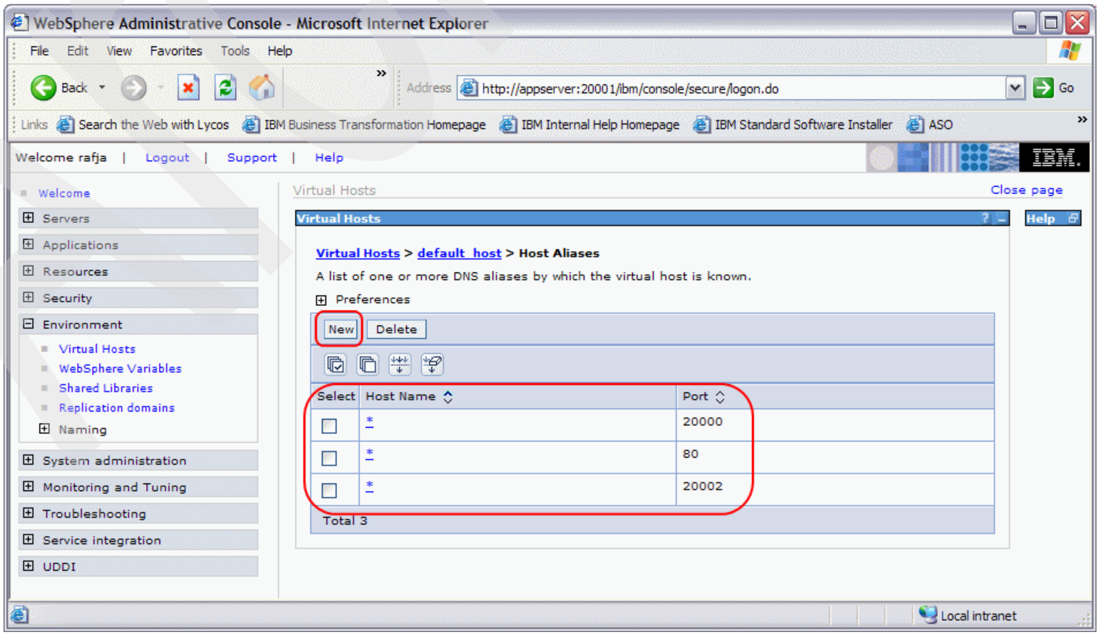
Configuration that lets a single host machine resemble multiple host machines. Each virtual host has a logical name and a list of one or more domain name system (DNS) aliases by which it is known.

Select	Name
□	admin_host
□	default_host
Total 2	

Figure 4-17 Virtual hosts list

6. The next page shows general and additional virtual host properties. under Additional Properties, click **Host Aliases**.

7. Figure 4-18 shows that there are three host aliases created. To associate the Web server instance with the application server profile, an entry that corresponds to the HTTP server instance port number must be in the Port column. In this scenario, the HTTP server instance uses port 80, as shown in Table 4-5 on page 72, and port 80 is one of the ports listed in Figure 4-18. This means that a new host alias entry does not need to be created for this scenario.


WebSphere Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://appserver:20001/ibm/console/secure/logon.do
Go
Links Search the Web with Lycos IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer ASO
Welcome rafja | Logout | Support | Help
IBM.
Welcome
Servers
Applications
Resources
Security
Environment
Virtual Hosts
WebSphere Variables
Shared Libraries
Replication domains
Naming
System administration
Monitoring and Tuning
Troubleshooting
Service integration
UDDI
Virtual Hosts
Close page
Help
Preferences
New
Delete
Local intranet

Virtual Hosts > default_host > Host Aliases

A list of one or more DNS aliases by which the virtual host is known.

Select	Host Name	Port
□	*	20000
□	*	80
□	*	20002
Total 3		

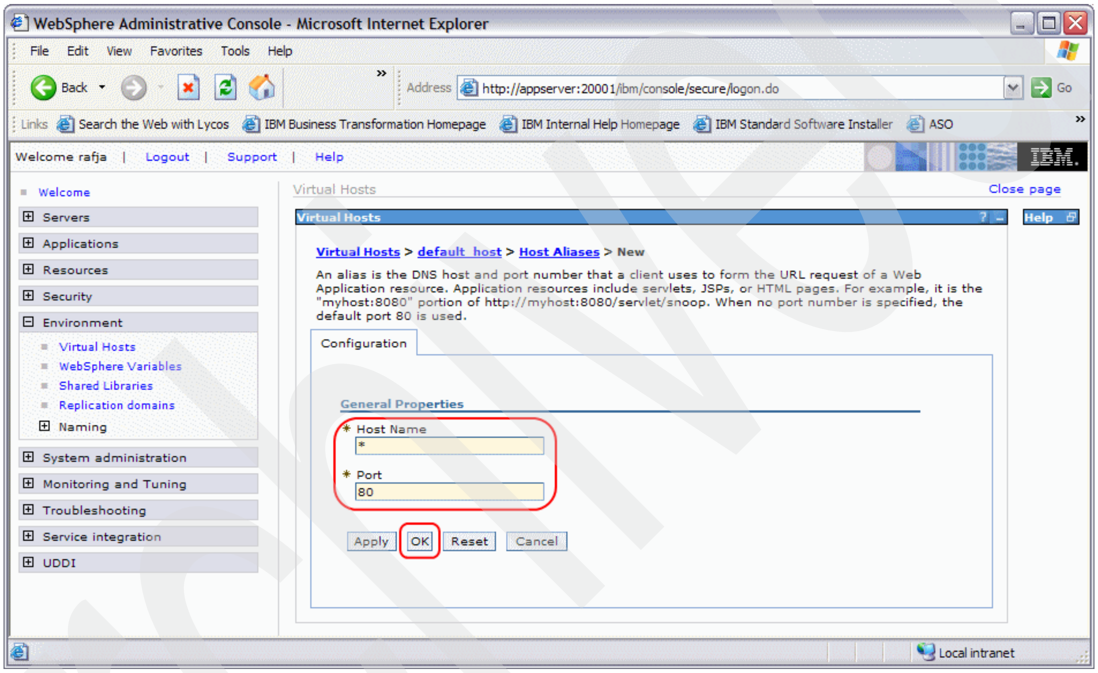
Figure 4-18 Virtual hosts aliases

Attention: A host alias with port 80 is created by default for all WebSphere Application Server profiles. If your HTTP Server instance uses a port different than 80, you must create a new host alias, with the port number that corresponds to the HTTP Server instance.

From a security point of view, you can narrow down which servers can access the WebSphere profile. Replace the asterisk (*) symbol with the host name of the server that is allowed to access this WebSphere profile.

To create a new host alias in the host aliases list, click the **New** button (see Figure 4-18).

8. In the next panel, complete the parameters as shown in Figure 4-19. Click **OK**.


WebSphere Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://appserver:20001/ibm/console/secure/logon.do
Go
Links Search the Web with Lycos IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer ASO
Welcome rafja | Logout | Support | Help
IBM.
Welcome
Servers
Applications
Resources
Security
Environment
Virtual Hosts
WebSphere Variables
Shared Libraries
Replication domains
Naming
System administration
Monitoring and Tuning
Troubleshooting
Service integration
UDDI
Virtual Hosts
Close page
Virtual Hosts
Help
Local intranet

Virtual Hosts > default_host > Host Aliases > New

An alias is the DNS host and port number that a client uses to form the URL request of a Web Application resource. Application resources include servlets, JSPs, or HTML pages. For example, it is the "myhost:8080" portion of http://myhost:8080/servlet/snoop. When no port number is specified, the default port 80 is used.

Configuration

General Properties

* Host Name: *

* Port: 80

Apply | OK | Reset | Cancel

Figure 4-19 Create virtual host alias

9. The new host alias is created. Save your changes.

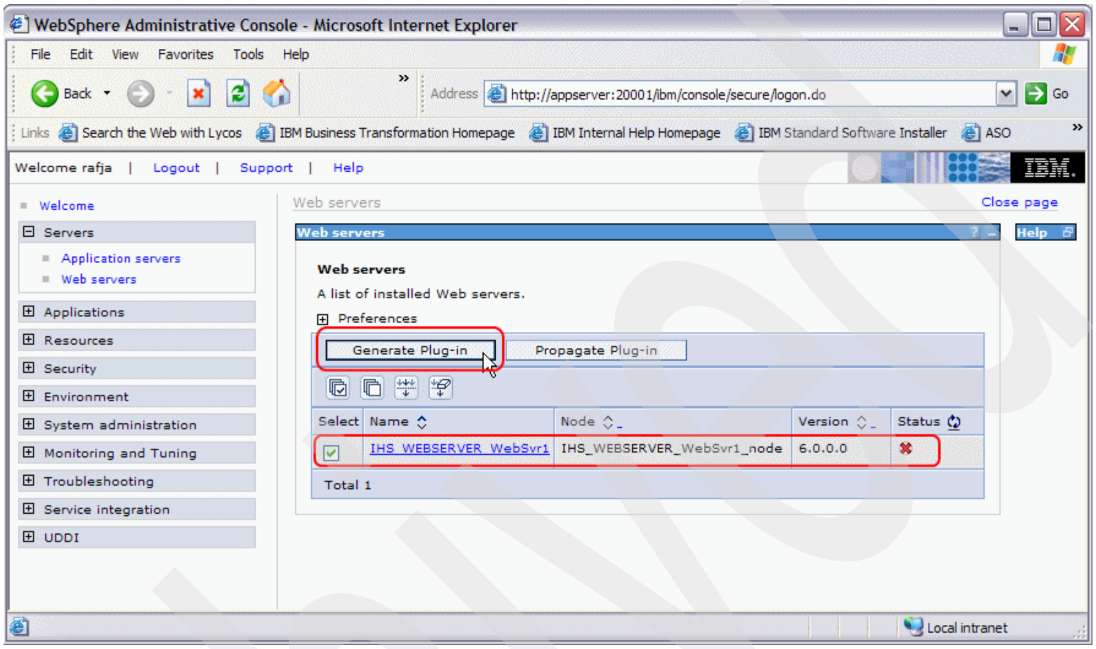
Important: Remember that all changes to the virtual hosts influence the plug-in configuration file. Regenerate the plug-in configuration file after each change.

Next you must generate the plugin-cfg.xml file which is then moved to the HTTP server iSeries host.

Generating plug-in configuration file

After a Web server definition and virtual hosts are created, the plug-in configuration file can be generated. The HTTP Server is located on a different iSeries from the application server profile. This means that the regenerated configuration file needs to be moved to the Web server iSeries.

1. Using the WebSphere Administrative Console on the AppServer iSeries, expand **Servers** and click **Web servers**.
2. To generate the plug-in configuration file, on the Web servers panel (Figure 4-20), in the Select column, select the check box next to the HTTP profile name. Click **Generate Plug-in**.


WebSphere Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://appserver:20001/ibm/console/secure/logon.do
Go
Links Search the Web with Lycos IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer ASO
Welcome rafja | Logout | Support | Help
IBM.
Welcome
Servers
Application servers
Web servers
Applications
Resources
Security
Environment
System administration
Monitoring and Tuning
Troubleshooting
Service integration
UDDI
Web servers
Close page
Help
Generate Plug-in
Propagate Plug-in
Local intranet

Web servers

A list of installed Web servers.

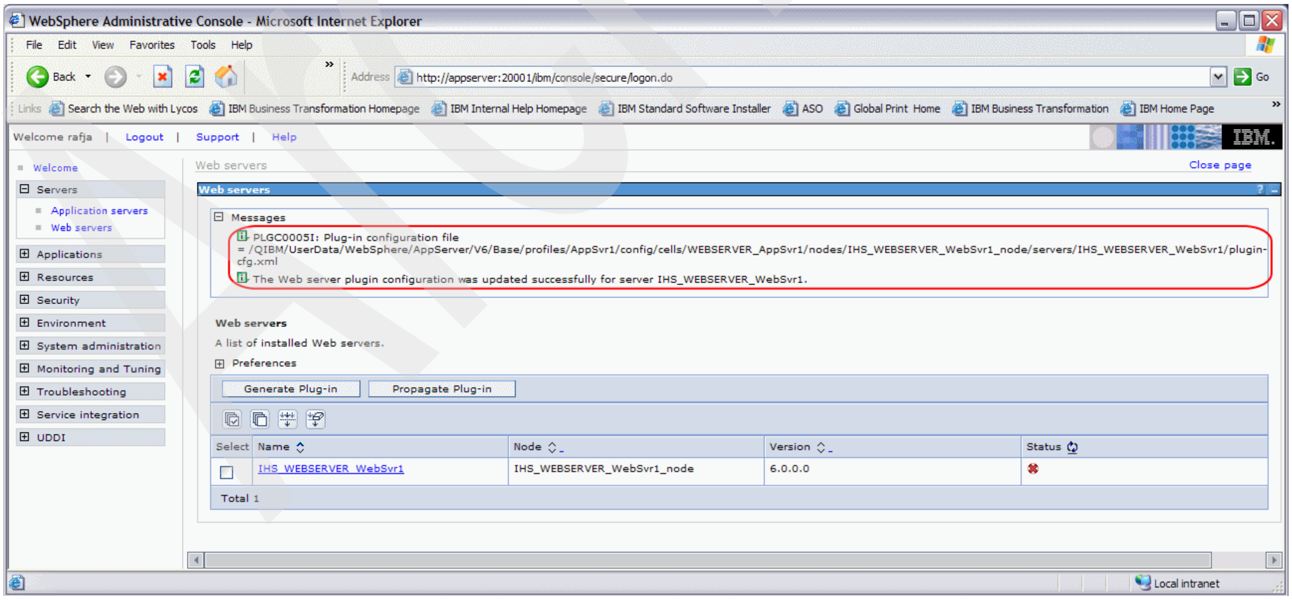
⊞ Preferences

Select	Name	Node	Version	Status
☑	IHS_WEBSERVER_WebSvr1	IHS_WEBSERVER_WebSvr1_node	6.0.0.0	✖

Total 1

Figure 4-20 Generating the plug-in configuration file

After the plug-in file is regenerated, you see a confirmation message and the location of the plug-in configuration file (see Figure 4-21).


WebSphere Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://appserver:20001/ibm/console/secure/logon.do
Go
Links Search the Web with Lycos IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer ASO Global Print Home IBM Business Transformation IBM Home Page
Welcome rafja | Logout | Support | Help
IBM.
Welcome
Servers
Application servers
Web servers
Applications
Resources
Security
Environment
System administration
Monitoring and Tuning
Troubleshooting
Service integration
UDDI
Web servers
Close page
Generate Plug-in
Propagate Plug-in
Local intranet

Messages

PLGC0005I: Plug-in configuration file = /QIBM/UserData/WebSphere/AppServer/V6/Base/profiles/AppSvr1/config/cells/WEBSERVER_AppSvr1/nodes/IHS_WEBSERVER_WebSvr1_node/servers/IHS_WEBSERVER_WebSvr1/plugin-cfg.xml

The Web server plugin configuration was updated successfully for server IHS_WEBSERVER_WebSvr1.

Web servers

A list of installed Web servers.

⊞ Preferences

Select	Name	Node	Version	Status
□	IHS_WEBSERVER_WebSvr1	IHS_WEBSERVER_WebSvr1_node	6.0.0.0	✖

Total 1

Figure 4-21 Generating the plug-in configuration file: Message confirmation

Important: Notice the location of the plug-in configuration file as shown in Figure 4-21. This information is required in future steps.

Generation of the plug-in configuration file is the final step required on the application server iSeries. The next activities are done on the HTTP server iSeries.

Creating an HTTP Server profile

This section demonstrates how to create a “dummy” HTTP server profile on the HTTP Server iSeries. The main reason to create an HTTP server profile is that, during the creation, an IFS directory structure, which is consistent with WebSphere Application Server directory structure, is also created. The HTTP Server instance looks for the plugin-cfg.xml file in this newly created directory structure. Log files related to the plug-in program are also generated within this directory structure.

One of two parameters that is needed to create an HTTP Server profile is the profile name. A good practice for naming this parameter is to use the same name that has been assigned to the application server profile. This parameter is shown in Table 4-6 on page 79 in the Profile Name column. As specified in this table, a value for this parameter is AppSvr1 and is used in this scenario.

Perform the following steps to create an HTTP server profile.

1. On the OS/400 command line, enter the `STRQSH` command. Make sure that you perform these steps on the HTTP server iSeries system.
2. On the Qshell command line, type the following command:

   ```
   cd /QIBM/ProdData/WebSphere/AppServer/V6/Base/bin
   ```

3. Run the wasprofile script. For this scenario, the command is:

   ```
   wasprofile -create -profileName AppSvr1 -templatePath http
   ```

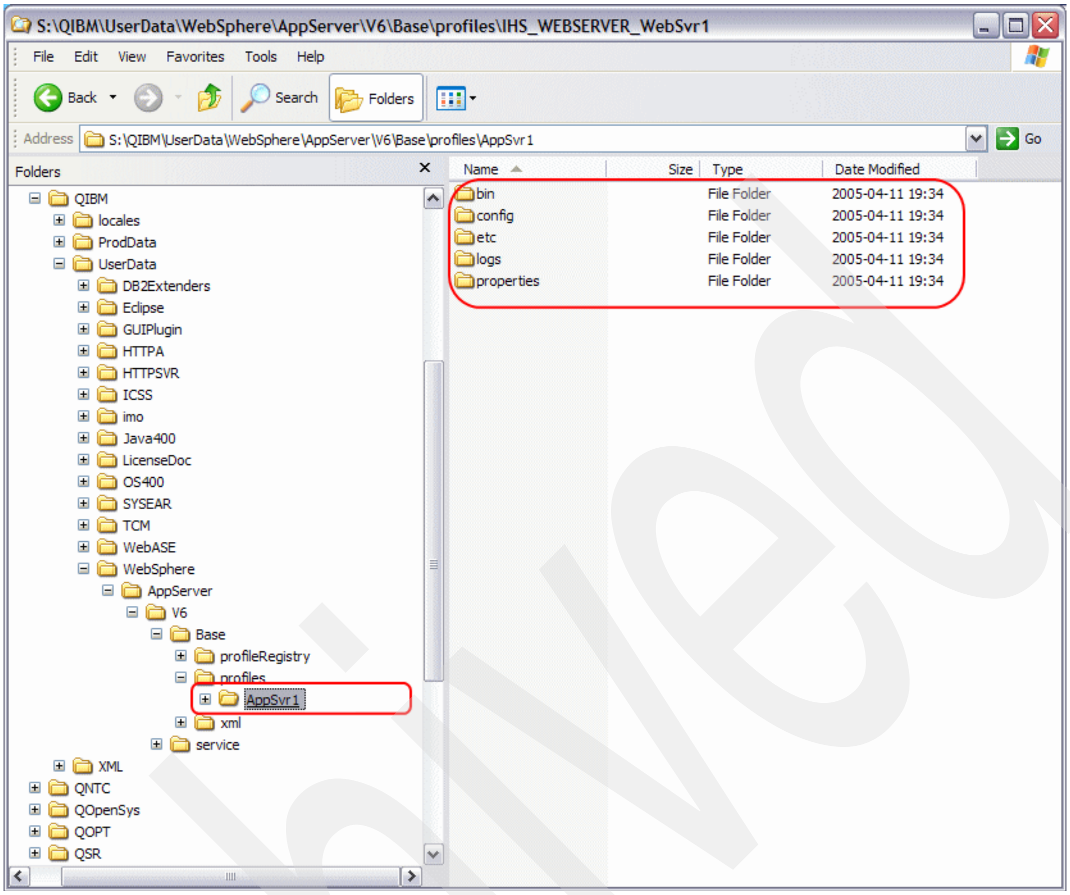
   Note: The templatePath parameter assumes that the specified template is in the *install_root*/profileTemplates directory. In this scenario, the http profile template is found in the /QIBM/ProdData/WebSphere/AppServer/V6/Base/profileTemplates directory.

   After the script is invoked, you see a message similar to the one in Example 4-13 in the Qshell session.

Example 4-13 Web server profile creation output

```
> wasprofile -profileName AppSvr1 -templatePath http
INSTCONFSUCCESS: Success: The profile now exists.
  $
```

The result of running the wasprofile script is the creation of a new directory structure as shown in Figure 4-22. The directory structure shown in Figure 4-22 is created within the /QIBM/UserData/WebSphere directory, which was originally created during the installation of the Web server plug-ins feature.


S:\QIBM\UserData\WebSphere\AppServer\V6\Base\profiles\IHS_WEBSERVER_WebSvr1

Address S:\QIBM\UserData\WebSphere\AppServer\V6\Base\profiles\AppSvr1

Name	Size	Type	Date Modified
bin		File Folder	2005-04-11 19:34
config		File Folder	2005-04-11 19:34
etc		File Folder	2005-04-11 19:34
logs		File Folder	2005-04-11 19:34
properties		File Folder	2005-04-11 19:34

Figure 4-22 Web server profile directory structure

Moving the plug-in configuration file

The application server iSeries generates the plug-in configuration file. However, the file is needed on the HTTP server iSeries specifically for the Web server instance. In this scenario, with the Web server and application server residing on different iSeries, the plug-in you need to move the configuration file from the application server iSeries to the Web server iSeries.

To successfully move the configuration file, you need to know its current location and the destination directory. The location is specified during the plug-in generation (see Figure 4-21 on page 87). The destination directory is on the Web server host. The destination directory is /QIBM/UserData/WebSphere/AppServer/V6/Base/profiles/*profileName*/config/, where *profileName* is the name of the HTTP profile created on the Web server iSeries. In this case, it has the same name as the WebSphere Application Server profile. Once you know the source and destination directories, you can move the plug-in configuration file to the Web server iSeries using any method such as FTP, a mapped drive, and so on.

Attention: After you move the plug-in configuration file to the Web server iSeries, check the authorities of this file. Run the following OS/400 command to check the current authorities of the plugin-cfg.xml file:

```
WRKAUT OBJ(‘QIBM/UserData/WebSphere/AppServer/V6/Base/profiles/profileName/
config/plugin-cfg.xml’)
```

Ensure the QTMHHTTP user has read (R) authority. If the user does not have this authority, then manually add it.

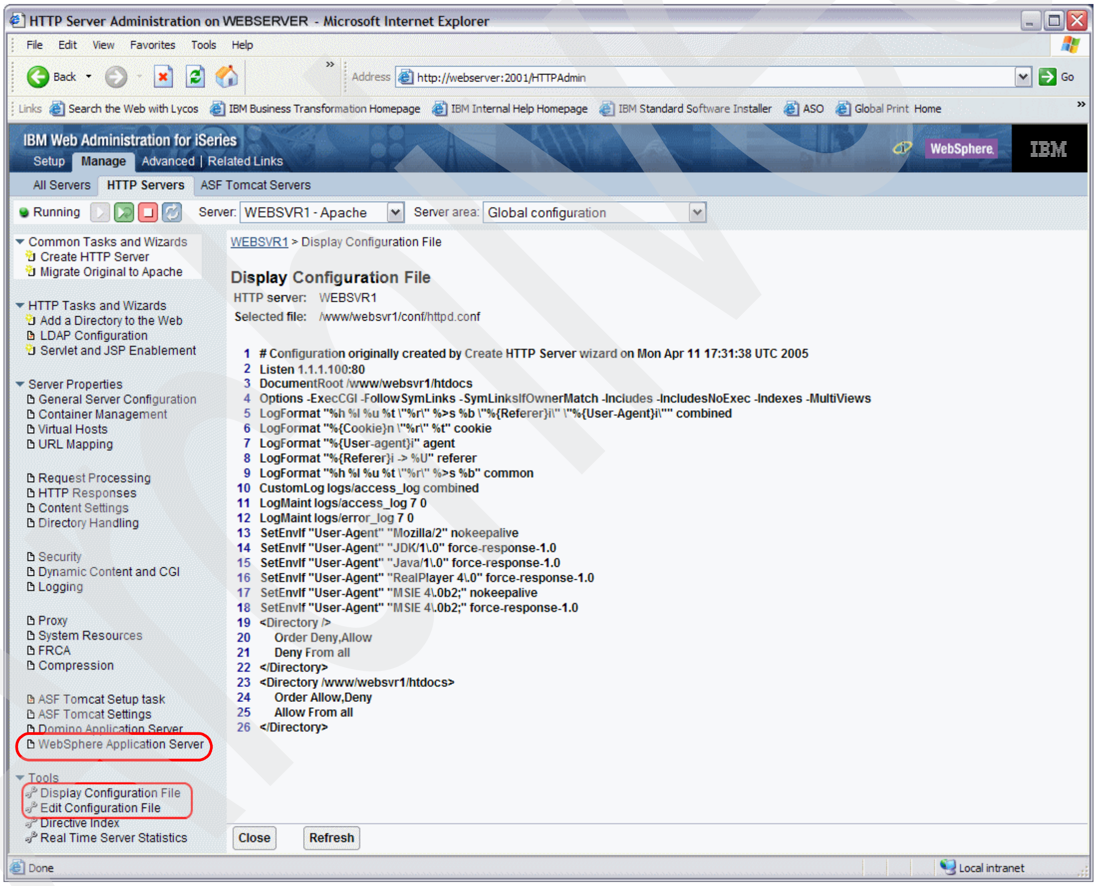
Configuring the HTTP Server plug-in

In this scenario, the application server works with the Web server to have information routed to and from Web applications. It is the responsibility of the Web server to:

- Direct traffic from client browsers to applications that run on the application server
- Route information from the application server back to the client network (Internet)

The Web server uses its plug-in component to support this communication. To establish communication between the Web server and the application server, the Web server needs to know where to find the plug-in service program and the plug-in configuration file. To configure the plug-in component, use Web Administration for iSeries.

1. Access the IBM Web Administration for iSeries GUI and log in. Refer to “Starting IBM Web Administration for iSeries” on page 68.
2. Click the **Manage** tab and click the HTTP server used in this configuration.
3. You see a page like the one in Figure 4-23. In the left navigation pane, click **WebSphere Application Server**.


HTTP Server Administration on WEBSERVER - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://webserver:2001/HTTPAdmin
Go
Links Search the Web with Lycos IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer ASO Global Print Home
IBM Web Administration for iSeries
Setup Manage Advanced | Related Links
WebSphere
IBM
All Servers HTTP Servers ASF Tomcat Servers
Running
Server: WEBSVR1 - Apache
Server area: Global configuration
Common Tasks and Wizards
Create HTTP Server
Migrate Original to Apache
HTTP Tasks and Wizards
Add a Directory to the Web
LDAP Configuration
Servlet and JSP Enablement
Server Properties
General Server Configuration
Container Management
Virtual Hosts
URL Mapping
Request Processing
HTTP Responses
Content Settings
Directory Handling
Security
Dynamic Content and CGI
Logging
Proxy
System Resources
FRCA
Compression
ASF Tomcat Setup task
ASF Tomcat Settings
Domino Application Server
WebSphere Application Server
Tools
Display Configuration File
Edit Configuration File
Directive Index
Real Time Server Statistics
WEBSVR1 > Display Configuration File
Close
Refresh
Done
Local intranet

Display Configuration File

HTTP server: WEBSVR1

Selected file: /www/websvr1/conf/httpd.conf

```
# Configuration originally created by Create HTTP Server wizard on Mon Apr 11 17:31:38 UTC 2005
Listen 1.1.1.100:80
DocumentRoot /www/websvr1/htdocs
Options -ExecCGI -FollowSymLinks -SymLinksIfOwnerMatch -Includes -IncludesNoExec -Indexes -MultiViews
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%{Cookie}n \"%r\" %t" cookie
LogFormat "%{User-agent}i" agent
LogFormat "%{Referer}i -> %U" referer
LogFormat "%h %l %u %t \"%r\" %>s %b" common
CustomLog logs/access_log combined
LogMaint logs/access_log 7 0
LogMaint logs/error_log 7 0
SetEnvIf "User-Agent" "Mozilla/2" nokeepalive
SetEnvIf "User-Agent" "JDK/1\.0" force-response-1.0
SetEnvIf "User-Agent" "Java/1\.0" force-response-1.0
SetEnvIf "User-Agent" "RealPlayer 4\.0" force-response-1.0
SetEnvIf "User-Agent" "MSIE 4\.0b2;" nokeepalive
SetEnvIf "User-Agent" "MSIE 4\.0b2;" force-response-1.0
<Directory />
   Order Deny,Allow
   Deny From all
</Directory>
<Directory /www/websvr1/htdocs>
   Order Allow,Deny
   Allow From all
</Directory>
```

Figure 4-23 HTTP Server instance configuration file

4. The WebSphere Application Server configuration pane opens.
   a. Select the radio button next to your version and edition of WebSphere Application Server. In this example, it is the **Base edition**.
   b. A drop-down field is displayed. Select your WebSphere Application Server profile name from the list. In our case, the name is **AppSvr1**.
   c. Click **OK**.

5. In the left navigation pane, click **Display Configuration File**.

   The first two lines in the configuration file should look similar to this example:

   ```
   WebSpherePluginConfig
   /QIBM/UserData/WebSphere/AppServer/V6/Base/profiles/profileName/config/plugin-cfg.xml
   LoadModule was_ap20_module /QSYS.LIB/QWAS6.LIB/QSVTAP20.SRVPGM
   ```

   Here *profileName* is the name of the HTTP Server profile you created in “Creating an HTTP Server profile” on page 88. The same profile name is found in Table 4-7 on page 82 in the WebSphere Profile Name column.

   The first line instructs the HTTP Server instance where to find the plug-in configuration file. The second line is an HTTP server command which loads the Web server plug-in to the HTTP Web server instance.

   Click **OK** to save the changes.

At this point, the HTTP Server instance can be started, and all the changes that have been made will be in effect.

4.3.7 Functional verification testing

To ensure that the Web server instance routes requests to the application server, you can run a simple test. From the client network (Internet), open a new Web browser window, and enter the URL which requests the Snoop servlet. The Snoop server is installed, by default, on the WebSphere Application Server profile.

This test is almost the same as the one done in “Starting the WebSphere Application Server profile” on page 80. The only difference is that instead of specifying the application server name, the Web server name or IP address is specified in the URL. In this scenario, we use the following URL, where 1.1.1.100 is the IP address of the WEBSERVER iSeries.

```
http://1.1.1.100/Snoop
```

The resulting Web page output is similar to what is shown in Figure 4-15 on page 82 where the Snoop servlet was tested on the APPSERVER iSeries. This test, which was initiated from the client network, confirms that the Snoop servlet request, which works on the application server iSeries, was routed correctly to the Web server iSeries.

Another more complex application that you can use to verify this topology is the Trade6 application. This application is used in benchmark testing and can simulate users who are accessing the sample topology that was created in this small company scenario. Trade6 was also used to expose any potential missing or incorrect configuration in both the protocol and domain firewalls. You can find more information about the Trade6 application in Appendix C, “Installing the Trade6 application” on page 277.

5

Scenario 2: Configuration for a large company

This chapter presents the components and outlines the setup steps required to implement a configuration for a large company. This chapter is a logical continuation of Chapter 4, “Scenario 1: Configuration for a small company” on page 61. Although the scenarios are presented in separate chapters, many of the concepts and network settings used in the configuration scenario for a small company carry through to the scenario for a large company.

Expanding the small company scenario, this chapter implements additional network components to create a more highly available and secure topology. The additional components include Lightweight Directory Access Protocol (LDAP) servers, load balancers, a second Web server, a WebSphere cluster, and a remote database system. In addition, a cross-site mirroring (XSM) channel is implemented for database backup and failover protection.

This chapter also presents the different components that make up the topology and the specific Transmission Control Protocol/Internet Protocol (TCP/IP) settings for each component. These components include the firewalls, the Web servers, application servers, load balancers, and LDAP servers. You can follow step-by-step instructions to help you set up all components of this topology.

Finally this chapter provides verification steps to help you ensure that the configuration is working correctly.

5.1 Large company configuration

In this chapter, the implementation of a topology, suitable for a large company, is discussed. The topology described in this chapter can be extended to any number of WebSphere and HTTP servers to satisfy the scalability goals.

5.1.1 Installed WebSphere software

For the large company scenario, WebSphere Application Server V6 software is installed on four of the iSeries systems. Refer to Table 5-1 for a listing of the edition and features that are installed on each iSeries.

Table 5-1 Installed software for the large company scenario

iSeries name	**WebSphere Application Server V6 edition**	**Installed features**
WEBSERVER1	Network Deployment	Web server plug-in only
WEBSERVER2	Network Deployment	Web server plug-in only
APPSERVER1	Network Deployment	All features
APPSERVER2	Network Deployment	All features
EdgePrimary	Network Deployment	Edge Components only
EdgeBackup	Network Deployment	Edge Components only

To install the software, refer to Appendix B, “Installing WebSphere Application Server Version 6” on page 255, and Appendix D, “Installing Load Balancer” on page 281. Before you continue with the next section, the WebSphere Application Server V6 software should be installed on each system.

5.2 Network configuration

Network infrastructure for computer systems is like the nervous system in the human body. It allows all the components to communicate with each other. You must think about the network infrastructure at the beginning because it influences steps later in the configuration. It is important from both the communication point of view and even more so from the security perspective. By isolating network devices in different subnetworks, you can achieve an appropriate level of security.

Figure 5-1 depicts the network topology that is used in the large customer scenario. As shown in Figure 5-1, there are three networks in this scenario. Each network is separated from the next network through a firewall. Each of these networks uses a different TCP/IP address range.

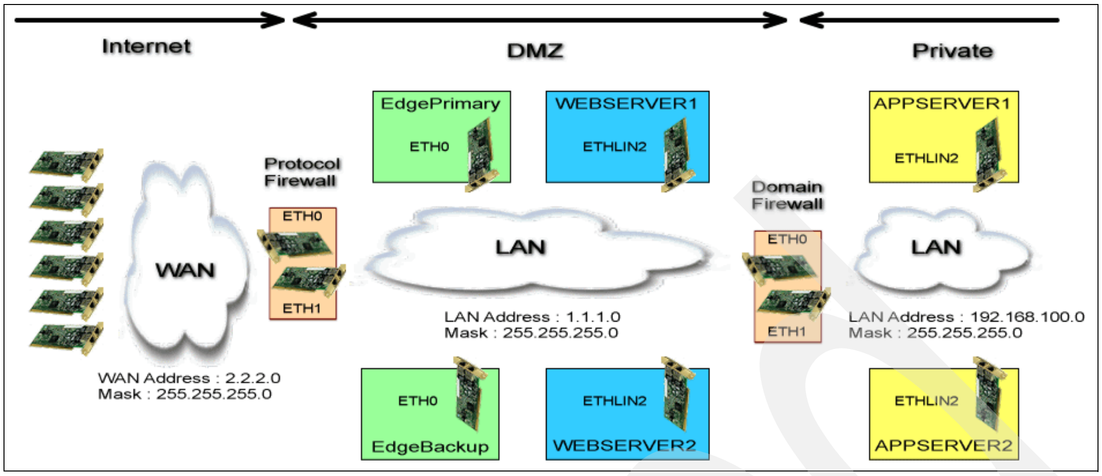


Figure 5-1 Network topology

The following networks make up the topology:

- **Private local area network (LAN)**: This network represents the internal company LAN. This is the most secured and protected network. The most important data is processed and stored in this network. No one from an outside network can reach this server directly. This private network contains two application servers. A database server resides on one of these application servers. And a backup database server exists in this network but is not shown in Figure 5-1.
- **Demilitarized zone (DMZ)**: This network represents the network between the two firewalls. Usually, all the Web servers, which serve static and less confidential data, are placed in this network. The DMZ has its own Internet Protocol (IP) address range, different from both the private and Internet networks. In this scenario, two HTTP Servers and two Edge Servers are configured in the DMZ.
- **Internet**: This network represents all potential Web users. It is the least secured network because potential hacker attacks come from this network. This network has a different address range from the other two networks. The protocol firewall is the first line of defense for the company intranet.

Table 5-2 shows the network settings for this scenario, including all TCP/IP address ranges and network resources.

Table 5-2 Network inventory

Network	Network address	Network mask	IP addresses range	Network adapters/server or device
Internet	2.2.2.0	255.255.255.0	2.2.2.1 - 2.2.2.254	* All Web users * ETH0/protocol firewall[a]
DMZ	1.1.1.0	255.255.255.0	1.1.1.1 - 1.1.1.254	* ETH1/protocol firewall[a] * ETH0/domain firewall[a] * ETH0/EdgePrimary * ETH0/EdgeBackup * ETHLIN2/WEBSERVER1 * ETHLIN2/WEBSERVER2
Private	192.168.100.0	255.255.255.0	192.168.100.1 - 192.168.100.254	* ETH1/domain firewall[a] * ETHLIN2/APPSERVER1 * ETHLIN2/APPSERVER2

a. Each firewall used in the scenario has two adapters named ETH0 and ETH1. Ensure that, when either adapter is referred to, the specific firewall is also identified. Refer to Figure 5-1 on page 95 for an overview diagram of the network used in this scenario.

Table 5-3 shows actual IP addresses that are used on each component in the scenario.

Table 5-3 IP address assignment

Network adapter/device	IP address	Mask
ETH0/protocol firewall[a]	2.2.2.1	255.255.255.0
ETH1/ protocol firewall[a]	1.1.1.1	255.255.255.0
Not applicable/ Edge Servers Cluster	1.1.1.100	255.255.255.0
ETH0/EdgePrimary server	1.1.1.150	255.255.255.0
ETH0/EdgeBakup server	1.1.1.151	255.255.255.0
ETHLIN2/WEBSERVER1	1.1.1.200	255.255.255.0
*VIRTUALIP/ WEBSERVER1	1.1.1.100	255.255.255.0
ETHLIN2/WEBSERVER2	1.1.1.201	255.255.255.0
*VIRTUALIP/ WEBSERVER2	1.1.1.100	255.255.255.0
ETH0/domain firewall[a]	1.1.1.253	255.255.255.0
ETH1/domain firewall[a]	192.168.100.1	255.255.255.0
ETHLIN2/APPSERVER1	192.168.100.100	255.255.255.0
ETHLIN2/APPSERVER1[b]	192.168.100.150	255.255.255.0
ETHLIN2/APPSERVER2	192.168.100.101	255.255.255.0
ETHLINE2/DBBACKUP	192.168.100.102	255.255.255.0
ETHLINE2/DBBACKUP[c]	192.168.100.150	255.255.255.0

a. Each firewall used in the scenario has two adapters named ETH0 and ETH1. Ensure that, when either adapter is referred to, the specific firewall is also identified. Refer to Figure 5-1 on page 95 for an illustration of the network used in this scenario.
b. This IP address is used for geographic mirroring configuration
c. This IP address is used for geographic mirroring configuration

To allow IP packets to flow between these three networks, you must set up routing information. Routing tables are used by TCP/IP service in all the cases when packets have to be sent to any other network or subnetwork. Table 5-4 shows the routing rules applied to the Web server and application server.

The first two entries define routing between the Web servers and the Internet hosts. Referring back to Figure 5-1 on page 95, it is possible to assume that these first two rules should be applied to the Edge servers, which are the first components to receive Internet requests. This assumption could be correct. However the way in which HTTP responses flow back, to the Internet user's browser, is different. The Web servers send responses directly to the Internet and not back to the Edge servers. This is why the Web servers need to be configured with the next hop, of the route, back to the users' hosts.

The next two entries in Table 5-4 also refer to the Web servers. The rules define how the Web servers can reach the application servers, which reside in the private LAN. The last two entries in Table 5-4 define the routing rules that allows the application servers to communicate with the Web servers. These rules allow packets to leave the private network to travel to the DMZ network.

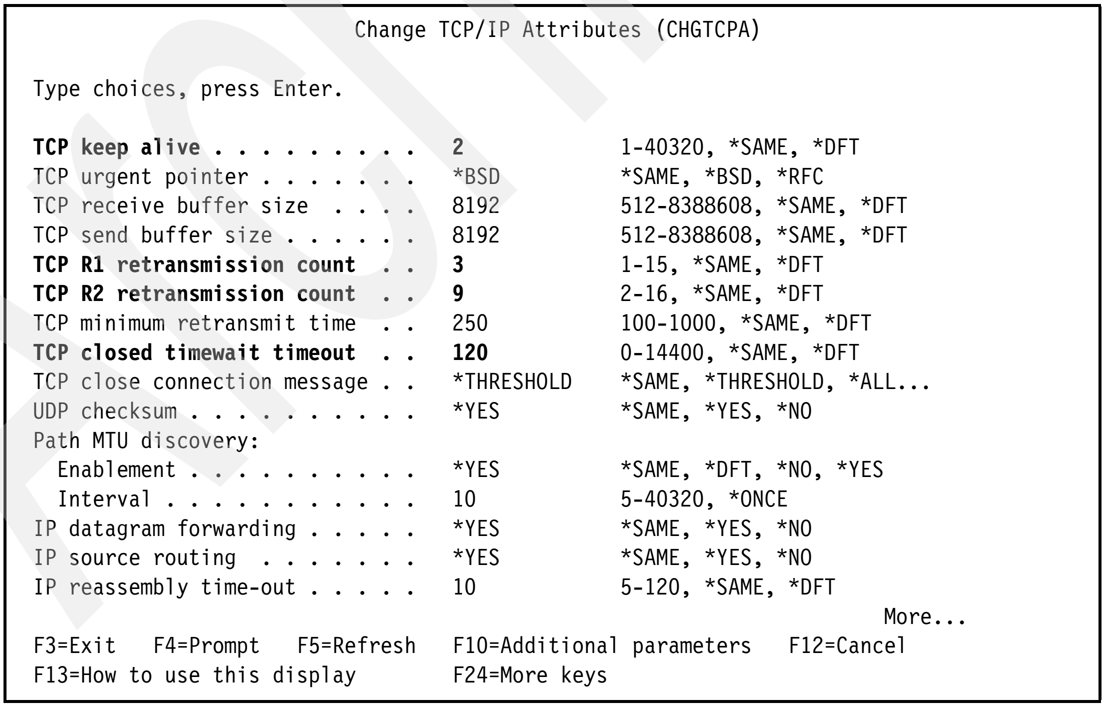
Table 5-4 Routing rules

Network adapter	Host name	Route destination	Mask	Next hop
ETHLIN2	WEBSERVER1	2.2.2.0 (Internet)	255.255.255.0	1.1.1.1(protocol firewall)
ETHLIN2	WEBSERVER2	2.2.2.0 (Internet)	255.255.255.0	1.1.1.1 (protocol firewall)
ETHLIN2	WEBSERVER2	192.168.100.0 (private LAN)	255.255.255.0	1.1.1.253 (domain firewall)
ETHLIN2	WEBSERVER2	192.168.100.0 (private LAN)	255.255.255.0	1.1.1.253 (domain firewall)
ETHLIN2	APPSERVER1	1.1.1.0 (DMZ)	255.255.255.0	192.168.100.1 (domain firewall)
ETHLIN2	APPSERVER2	1.1.1.0 (DMZ)	255.255.255.0	192.168.100.1 (domain firewall)

5.2.1 Configuring the TCP/IP settings on the iSeries

You should also configure some TCP/IP settings for the host, using the Change TCP/IP Attributes (CHGTCPA) command, to achieve a timely failover for your database and connections. Specifically verify the following attributes as shown in Figure 5-2:

- TCP keep alive: 2 minutes
- TCP R1 retransmission count: 3
- TCP R2 retransmission count: 9
- ARP cache timeout: 2 minutes



```
                              Change TCP/IP Attributes (CHGTCPA)

 Type choices, press Enter.

 TCP keep alive . . . . . . . . .   2             1-40320, *SAME, *DFT
 TCP urgent pointer . . . . . . .   *BSD          *SAME, *BSD, *RFC
 TCP receive buffer size  . . . .   8192          512-8388608, *SAME, *DFT
 TCP send buffer size . . . . . .   8192          512-8388608, *SAME, *DFT
 TCP R1 retransmission count  . .   3             1-15, *SAME, *DFT
 TCP R2 retransmission count  . .   9             2-16, *SAME, *DFT
 TCP minimum retransmit time  . .   250           100-1000, *SAME, *DFT
 TCP closed timewait timeout  . .   120           0-14400, *SAME, *DFT
 TCP close connection message . .   *THRESHOLD    *SAME, *THRESHOLD, *ALL...
 UDP checksum . . . . . . . . . .   *YES          *SAME, *YES, *NO
 Path MTU discovery:
   Enablement . . . . . . . . . .   *YES          *SAME, *DFT, *NO, *YES
   Interval . . . . . . . . . . .   10            5-40320, *ONCE
 IP datagram forwarding . . . . .   *YES          *SAME, *YES, *NO
 IP source routing  . . . . . . .   *YES          *SAME, *YES, *NO
 IP reassembly time-out . . . . .   10            5-120, *SAME, *DFT
                                                                        More...
 F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
 F13=How to use this display        F24=More keys
```

Figure 5-2 Setting TCP/IP attributes

Although you may need to make some additional adjustments for each specific configuration, these settings are a good starting point to optimize the timeouts to handle a database failover.

Important: All iSeries servers in the configuration should have these TCP attributes set to those values to ensure high availability (HA) characteristics for failover processing.

5.2.2 Detailed topology diagram

Figure 5-3 shows the entire topology, with IP settings, that we setup.

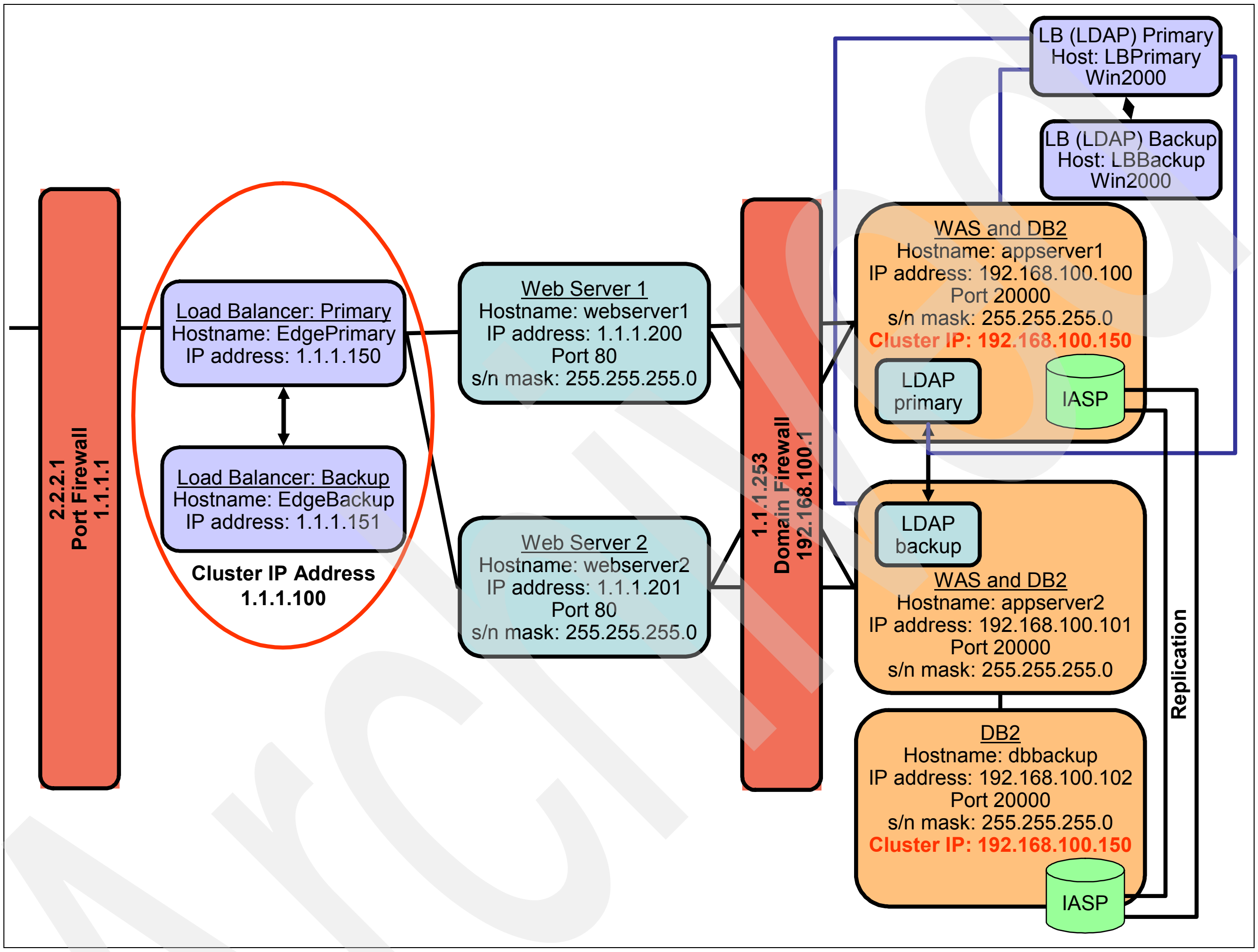


Figure 5-3 Large company configuration

In addition to WebSphere and HTTP servers, we have the following components:

- Two independent auxiliary storage pools (IASPs) on APPSERVER1 and DBBACKUP systems

  We use XSM technology to provide the HA option for the database.

- Two LDAP servers (primary and backup) on systems APPSERVER1 and APPSERVER2

  If your application uses LDAP for security, make sure it's highly available. To do that, you need to maintain two copies of LDAP data by replicating the primary server to the backup (replica). However, the LDAP server on iSeries doesn't have any built-in support for failover. To bypass this limitation, we implement the second pair of load balancers in front of LDAPs: LBPrimary and LBBackup. WebSphere servers access LDAP through a load balancer, which can switch to the replica copy of the LDAP data, if the primary LDAP server is not available.

In the remainder of this chapter, you learn how to configure this topology. You begin with the back end (the database) because the planning and setup require major system changes. It’s easier to do it while the system or systems are not configured with more software products or applications.

5.3 Data considerations for high availability

Access to enterprise data is critical for most e-business applications. Since many of those applications must be available around the clock, the associated data must also be highly available. In this section, you learn what you can do to make data highly available, including using IASPs and replication technologies.

The topology that we are working with in our HA complex configuration uses independent disk pools and XSM. There are various solutions for data high availability, but we focus only on XSM in this example. This section takes you through the steps necessary to implement XSM and to configure the TRADE6 enterprise Web application to work in that environment.

> **Note:** The example does not provide fast switchover in the case of a primary node failure. The IASP on the backup cluster node must be varied on, as should the takeover IP interface. This process can take four minutes or more to complete.

The WebSphere application example that we are using is TRADE6, which is an enterprise application that accesses a TRADE database for application data. To illustrate data considerations for high availability, we implement a solution using IASPs and XSM. In this configuration, we create a cluster (ITSOCLU) with two cluster nodes (ITSOSVR1, ITSOSVR2), which have IASPs (TRADE) on them with the TRADE database. Data replication through the cluster services will keep the database synchronized.

One common IP address will be used to address the cluster (takeover IP address 192.168.100.150). This IP address will be used for database access within the TRADE6 WebSphere enterprise application. See Figure 5-4 for details about the IASP and cluster layout. For this to work, we need to define additional communication cards and IP addresses, which are different from those used in Figure 5-3.

We refer to the cluster nodes as ITSOSVR1 and ITSOSVR2. These names correspond to the APPSERVER1 and DBBACKUP systems respectively.

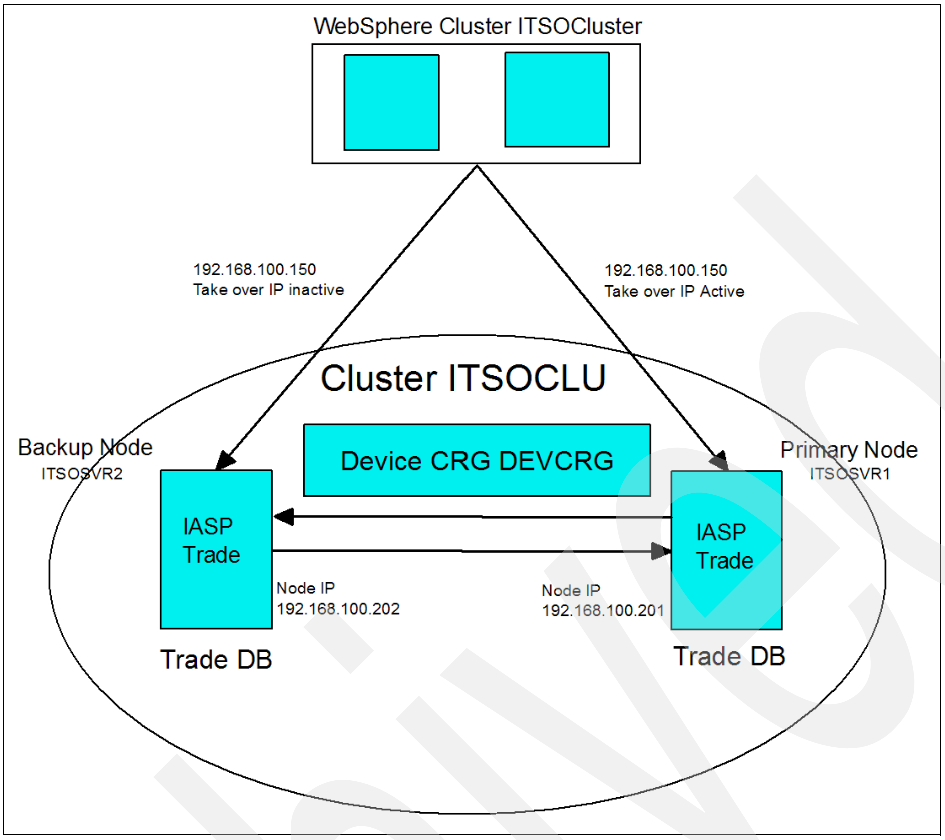


Figure 5-4 Data Replication Cluster ITSOCLU Topology

For detailed information, see *Clustering and IASPs for Higher Availability on the IBM @server iSeries Server*, SG24-5194. You can also find a variety of information about XSM and clusters in the iSeries Information Center at:

http://publib.boulder.ibm.com/infocenter/iseries/v5r3/index.jsp

Here is an overview of the tasks that you must follow to create the XSM with dedicated IASPs:

1. Create an IASP (Trade) on the primary system that is to be mirrored.
2. Create the cluster ITSOCLU.
3. Create a recovery domain for the nodes in the cluster.
4. Add nodes to the device domain.
5. Create the cluster resource group (CRG) device.
   a. Define the geographic mirroring sites in the recovery domain (ITSOSVR1, ITSOSVR2).
   b. Identify the IP addresses to be used by dataPort services.
6. Configure geographic mirroring under iSeries Navigator.
   a. Use a wizard to do the configuration.
   b. Identify the IASPs to include.
   c. Set the attributes and tuning parameters.

5.3.1 Creating the independent disk pool Trade

The first step in implementing XSM is to create an IASP on the node which will be the primary node in the cluster.

An independent disk pool is a collection of disk units that can be brought online or taken offline, independently of the rest of the storage on a system, including the system disk pool, basic user disk pools, and other independent disk pools.

An independent disk pool can be either switchable among multiple systems in a clustered environment or privately connected to a single system. The benefits in both multisystem clustered environments and single-system environments can be significant. For example, in a clustered environment, the use of independent disk pools can provide disk storage that is switchable among servers in the cluster, providing high availability of the resources. In a single-system environment, independent disk pools can be used to isolate infrequently used data that does not always need to be present when the system is operational.

With XSM, you extend the advantages of a stand-alone IASP. XSM allows you to use disks that are not capable of switching. For example, they are connected to the same input/output processor (IOP) as the system auxiliary storage pool (ASP). Yet these disks still provide a switching capability between the primary and backup copies of your data.

In our current environment, we use system ITSOSVR1 as the primary node and ITSOSVR2 as the backup node. This means that the IASP, which is going to house the TRADE database, will be the active database that the Web application initially accesses.

You configure and manage an IASP using the iSeries Navigator interface. You can use a “green-screen” interface to perform some administrative actions because IASP has been configured with the iSeries Navigator. This section highlights the aspects of creating an IASP using iSeries Navigator. You can find step-by-step instructions for creating the IASPs, using iSeries Navigator, in Chapter 7 of the IBM Redbook *Clustering and IASPs for Higher Availability on the IBM @server iSeries Server*, SG24-5194.

Access to the disk functions is controlled via the service tools user profiles. You must have access to a user profile authorized for System Service Tools (SST) before you continue.

Preparing the disk units

To configure an IASP on the system, you must have one or more nonconfigured disks. For this example, we remove one of the disk units from the system disk pool on both systems (ITSOSVR1 and ITSOSVR2). We also make sure that the size of this disk unit is the same on both systems (ITSOSVR1 and ITSOSVR2).

1. Restart the system in manual mode to the Dedicated System Tools (DST).
2. When the first screen appears, select option 3 (Use Dedicated Service Tools (DST)).
3. Sign in with the DST user ID and password.
4. Select option 4 (Work with disk units).
5. On the next display, select option 1 (Work with disk configuration).
6. On the next display, select option 3 (Work with ASP configuration).
7. On the next display, select option 7 (Remove units from configuration).
8. You should see the list of all disk units on the system. Type 4 on the option line against the disk unit that you want to remove and press Enter.
9. Press Enter again to confirm your action.
10. The system starts moving data from the disk unit that you selected to be removed. This can take a long time.
11. Perform an initial program load (IPL) of the system.

Now you have one or more disk units that you can use for configuring an IASP.

Creating the IASP

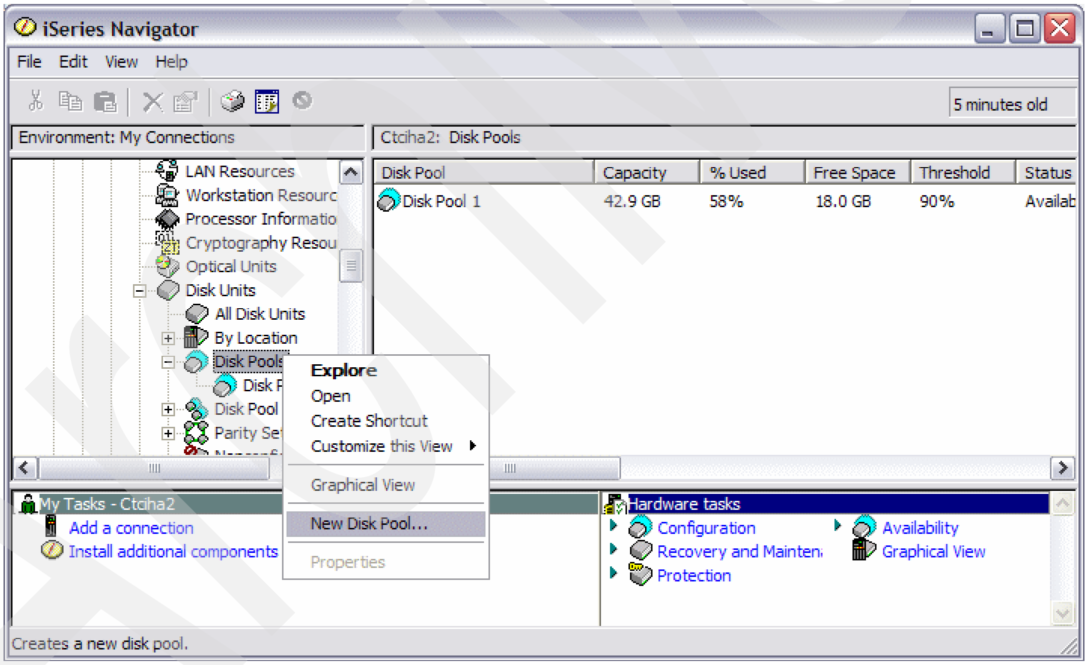
Now, when we have nonconfigured disk units, we can create the IASP. Having two IASPs is a required condition for configuring XSM.

> **Note:** While configuring XSM, you need to work with two or more iSeries systems. Depending on the step, you may need to sign on to a different system. The signon window is displayed, but may be hidden behind other open windows. For this reason, close or minimize all open windows before you perform the configuration steps.

To create the IASP, follow these steps:

1. Start iSeries Navigator.
2. Expand ***your system*** → **Configuration and Service** → **Hardware** → **Disk Units**. Right-click **Disk Pools** and select **New Disk Pool** (see Figure 5-5).

   > **Note:** You need to use the service tools user ID. To learn more about the service tools user profile, select the topics **Security** → **Service tools user IDs and passwords** in the iSeries Information Center at:
   >
   > `http://publib.boulder.ibm.com/infocenter/iseries/v5r3/index.jsp`



Disk Pool	Capacity	% Used	Free Space	Threshold	Status
Disk Pool 1	42.9 GB	58%	18.0 GB	90%	Availab

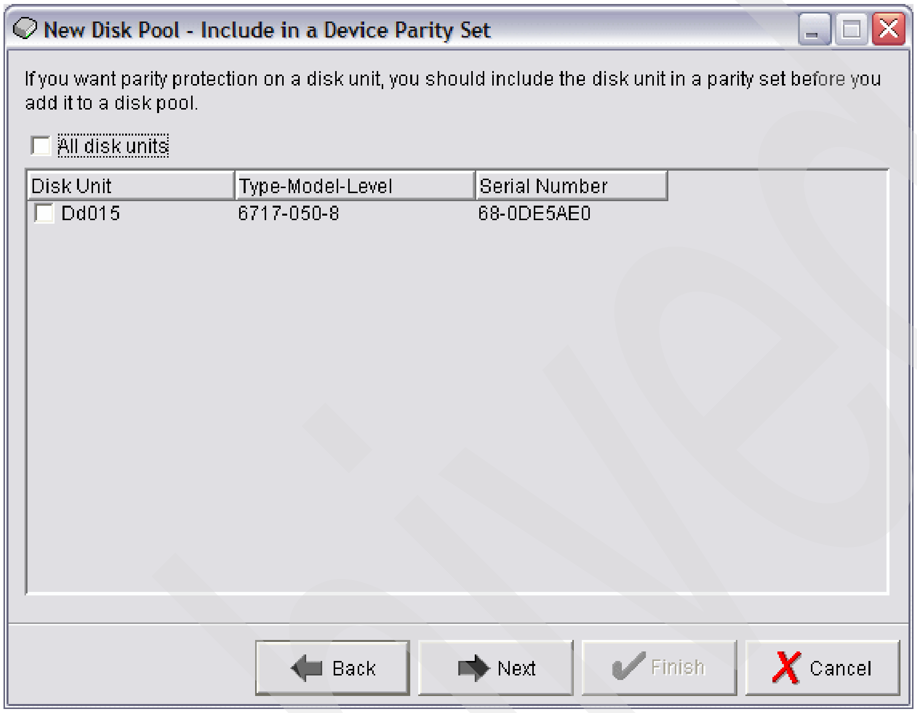
Figure 5-5 Creating a new disk pool

3. The New Disk Pool wizards starts. On the first panel, click **Next**.

   Prior to creating the disk pool, ensure that you have enough disk space and similar amounts of disk size available on both systems that you want to use to implement XSM.

4. Determine if you want parity protection. In the Include in a Device Parity Set panel (Figure 5-6), you can assign disk parity to your disks that you can add later to the IASP. If the disk is not protected, you do not see this panel.

   In our example, we do not use disk parity. However, we strongly recommend that you enable disk parity or i5/OS disk mirroring when you create your disk pools. Click **Next**.



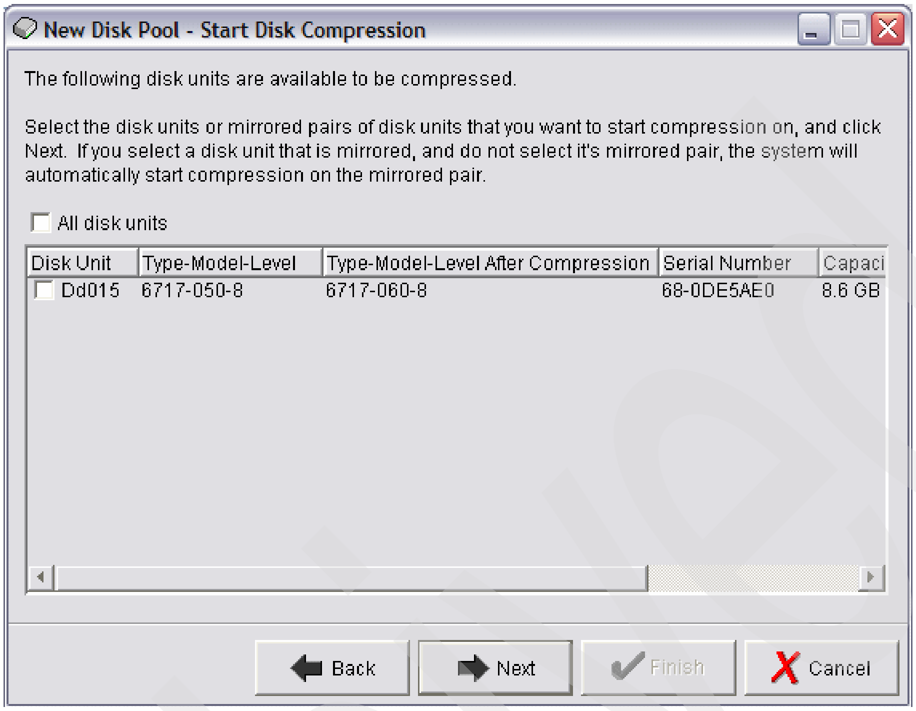
If you want parity protection on a disk unit, you should include the disk unit in a parity set before you add it to a disk pool.

□ All disk units

Disk Unit	Type-Model-Level	Serial Number
□ Dd015	6717-050-8	68-0DE5AE0

Figure 5-6 Available nonconfigured disk unit

5. In the Start Disk Compression panel (Figure 5-7), do not select the options that are shown, because compression is not optimal for production. Click **Next**.



The following disk units are available to be compressed.

Select the disk units or mirrored pairs of disk units that you want to start compression on, and click Next. If you select a disk unit that is mirrored, and do not select it's mirrored pair, the system will automatically start compression on the mirrored pair.

□ All disk units

Disk Unit	Type-Model-Level	Type-Model-Level After Compression	Serial Number	Capaci
□ Dd015	6717-050-8	6717-060-8	68-0DE5AE0	8.6 GB

Figure 5-7 Start Disk Compression panel

6. In the New Disk Pool window (Figure 5-8), complete the following actions.

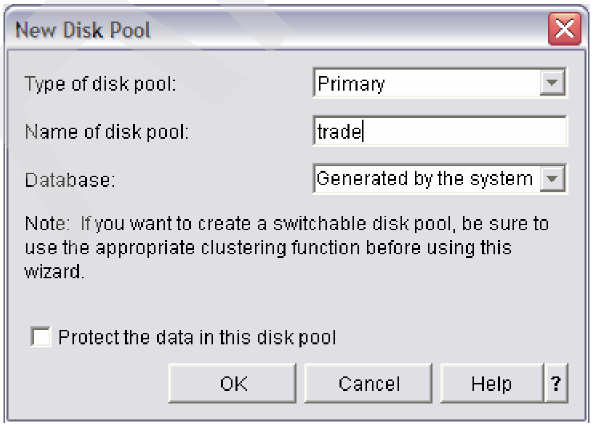
   a. From the Type of disk pool list, choose **Primary**.

   b. In the Name of disk pool field, type a name for the disk pool.

   c. For Database, you can type a name for the pool or accept the default value. The wizard creates the name to be the same as the IASP name. In this case, a new WRKRDBDIRE entry will be added for TRADE on system ITSOSVR1. This allows access to the IASP on that system.

   d. Deselect **Protect the data in this disk pool**. Use this option only if you plan to protect the data in this disk pool with mirroring (this is not XSM) or device parity protection.

   e. Click **OK**.



Type of disk pool: Primary

Name of disk pool: trade

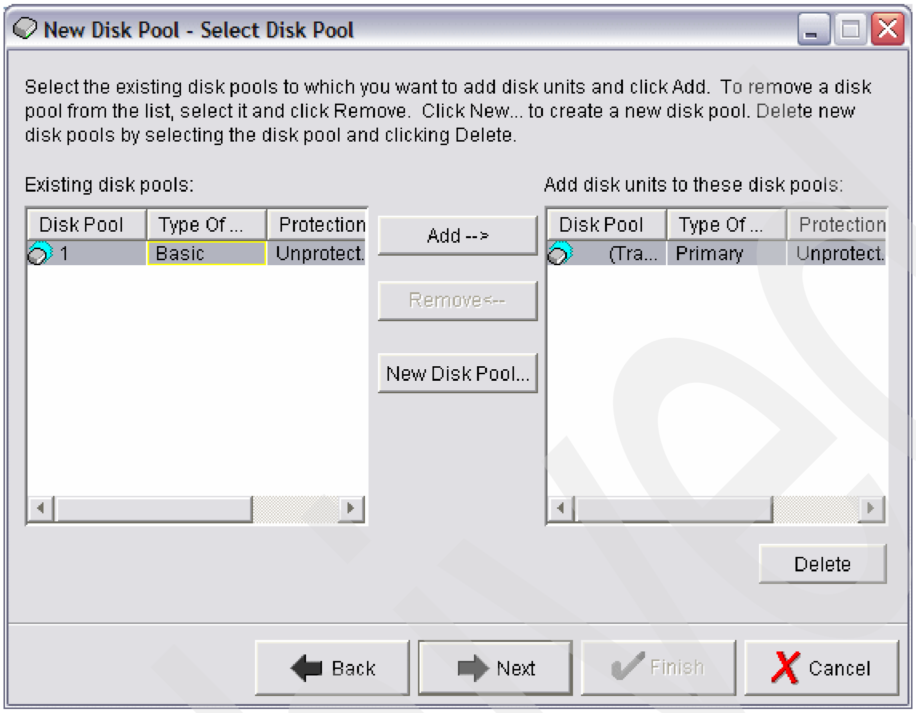
Database: Generated by the system

Note: If you want to create a switchable disk pool, be sure to use the appropriate clustering function before using this wizard.

□ Protect the data in this disk pool

Figure 5-8 Selecting the type and the name for the disk pool

7. The Select Disk Pool panel (Figure 5-9) opens. This particular panel allows you to configure more disk pools if necessary. The disk pool that is listed is the only disk pool that we configure for this example. Click **Next**.



New Disk Pool - Select Disk Pool

Select the existing disk pools to which you want to add disk units and click Add. To remove a disk pool from the list, select it and click Remove. Click New... to create a new disk pool. Delete new disk pools by selecting the disk pool and clicking Delete.

Existing disk pools:

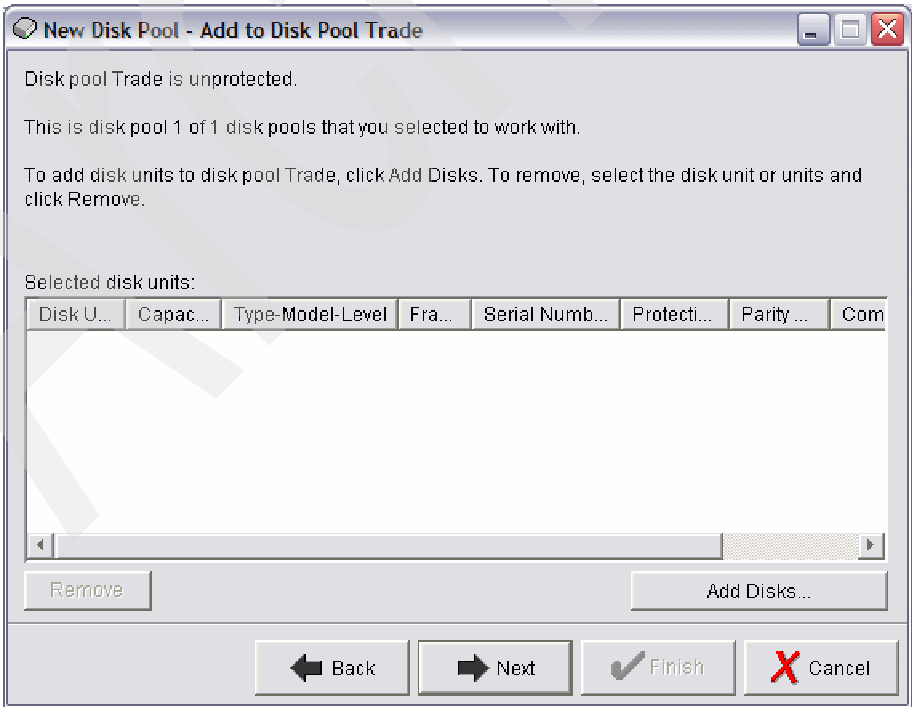
Disk Pool	Type Of ...	Protection
1	Basic	Unprotect.

Add disk units to these disk pools:

Disk Pool	Type Of ...	Protection
(Tra...	Primary	Unprotect.

Figure 5-9 New Disk Pool - Select Disk Pool panel

8. The Add to Disk Pool panel (Figure 5-10) opens. Now that we have defined a disk pool, add a non-configured disk to the pool. Click **Add Disks** to add disks to the pool.



New Disk Pool - Add to Disk Pool Trade

Disk pool Trade is unprotected.

This is disk pool 1 of 1 disk pools that you selected to work with.

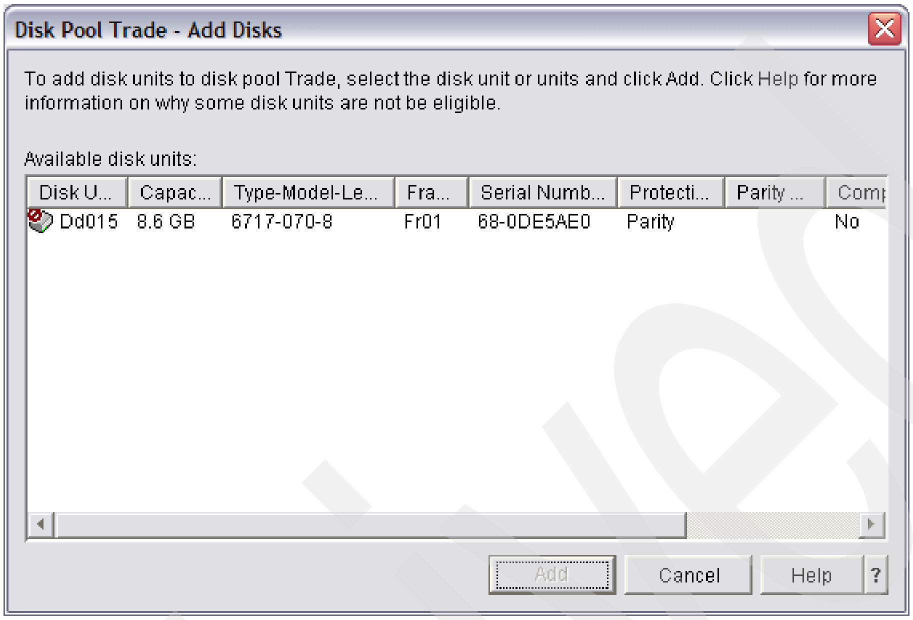
To add disk units to disk pool Trade, click Add Disks. To remove, select the disk unit or units and click Remove.

Selected disk units:

Disk U...	Capac...	Type-Model-Level	Fra...	Serial Numb...	Protecti...	Parity ...	Com

Figure 5-10 Adding disk to the pool

a. In the Add Disks window (Figure 5-11), you see the nonconfigured disks on the system. Before we start the configuration of the IASP, you must have a nonconfigured disk available for the disk pool. Select the disk that you want to use and click **Add** to add it to the disk pool.



Disk Pool Trade - Add Disks

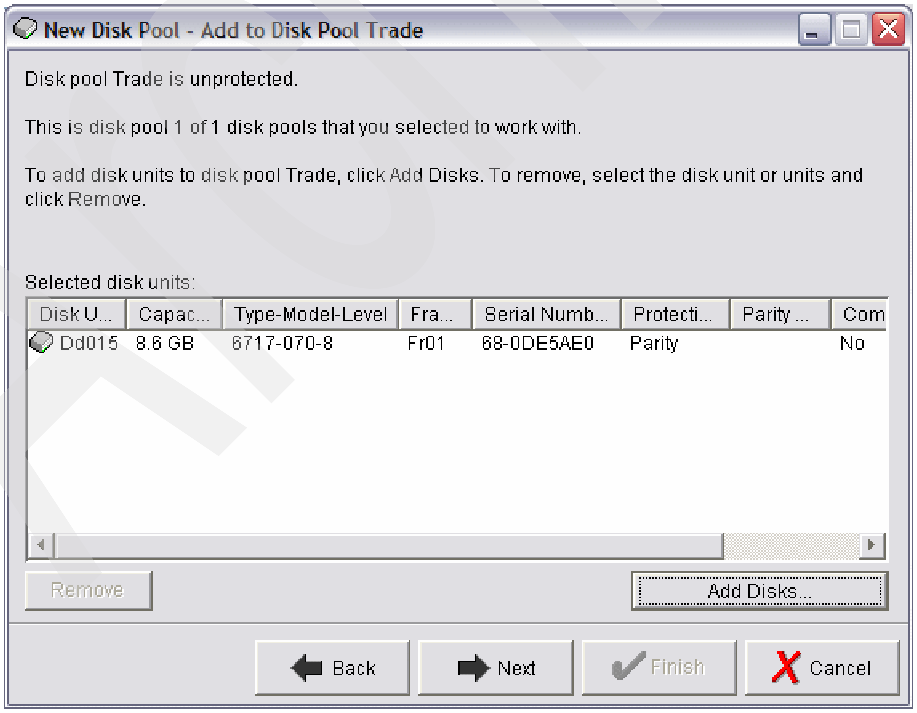
To add disk units to disk pool Trade, select the disk unit or units and click Add. Click Help for more information on why some disk units are not be eligible.

Available disk units:

Disk U...	Capac...	Type-Model-Le...	Fra...	Serial Numb...	Protecti...	Parity ...	Comp
Dd015	8.6 GB	6717-070-8	Fr01	68-0DE5AE0	Parity		No

Figure 5-11 Selecting a nonconfigured disk to add to the newly created disk pool

b. Back in the Add to Disk Pool panel (Figure 5-12), verify that the disk unit is added to the disk pool and that the related details are correct. Click **Next**.



New Disk Pool - Add to Disk Pool Trade

Disk pool Trade is unprotected.

This is disk pool 1 of 1 disk pools that you selected to work with.

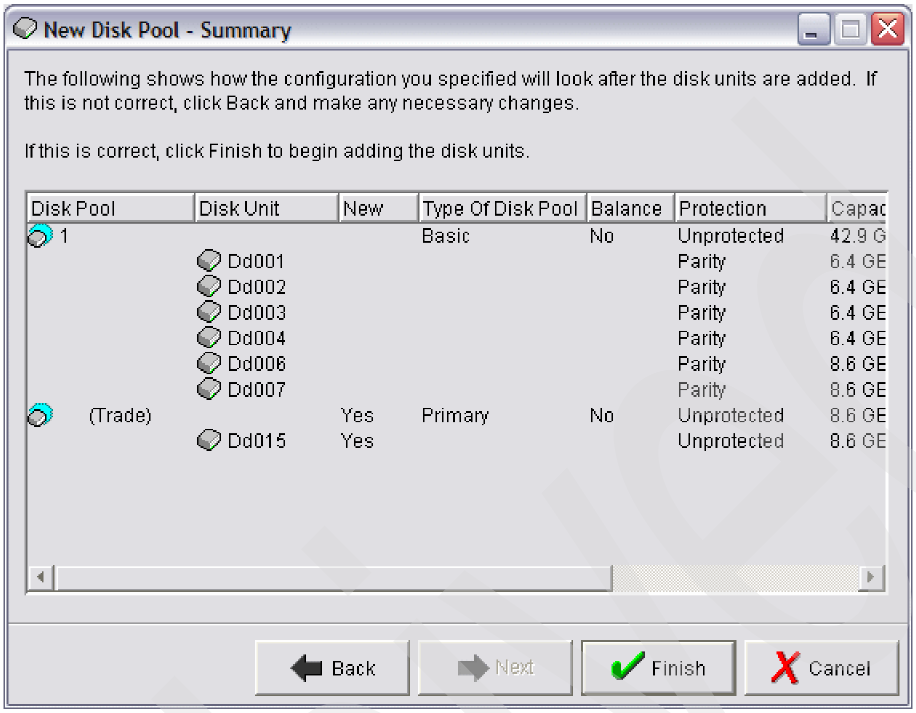
To add disk units to disk pool Trade, click Add Disks. To remove, select the disk unit or units and click Remove.

Selected disk units:

Disk U...	Capac...	Type-Model-Level	Fra...	Serial Numb...	Protecti...	Parity ...	Com
Dd015	8.6 GB	6717-070-8	Fr01	68-0DE5AE0	Parity		No

Figure 5-12 Confirming that a disk unit was added to the disk pool

9. In the Summary panel (Figure 5-13), verify the disk pool configuration. Click **Finish** to create the pool.



New Disk Pool - Summary

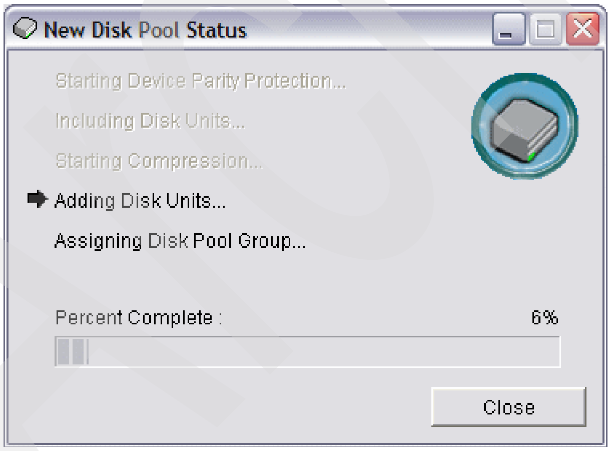
The following shows how the configuration you specified will look after the disk units are added. If this is not correct, click Back and make any necessary changes.

If this is correct, click Finish to begin adding the disk units.

Disk Pool	Disk Unit	New	Type Of Disk Pool	Balance	Protection	Capac
1			Basic	No	Unprotected	42.9 G
	Dd001				Parity	6.4 GE
	Dd002				Parity	6.4 GE
	Dd003				Parity	6.4 GE
	Dd004				Parity	6.4 GE
	Dd006				Parity	8.6 GE
	Dd007				Parity	8.6 GE
(Trade)		Yes	Primary	No	Unprotected	8.6 GE
	Dd015	Yes			Unprotected	8.6 GE

Figure 5-13 Summary of the disk pool configuration

10. The New Disk Pool Status window (Figure 5-14) opens to indicate that the disk pool is now in the process of being created.



New Disk Pool Status

Starting Device Parity Protection...

Including Disk Units...

Starting Compression...

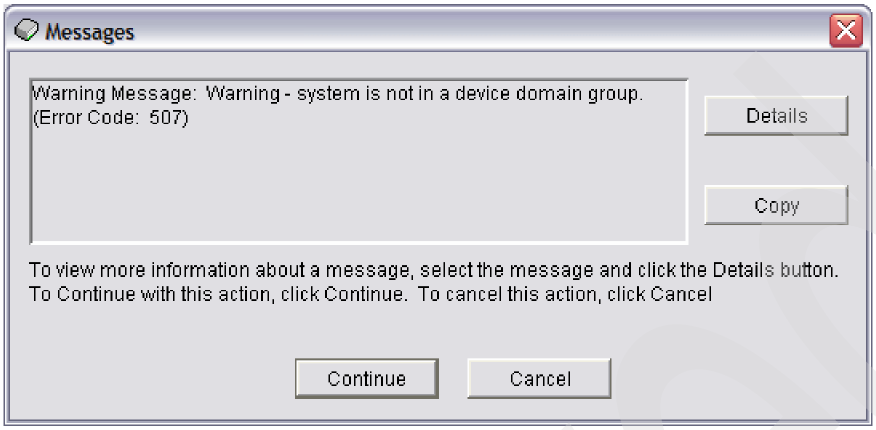
➡ Adding Disk Units...

Assigning Disk Pool Group...

Percent Complete : 6%

Figure 5-14 Disk pool creation progress

11. After creation of the disk pool is complete, you see a warning message indicating that the system is not in a device domain group (Figure 5-15). This is acceptable, so click **Continue**.

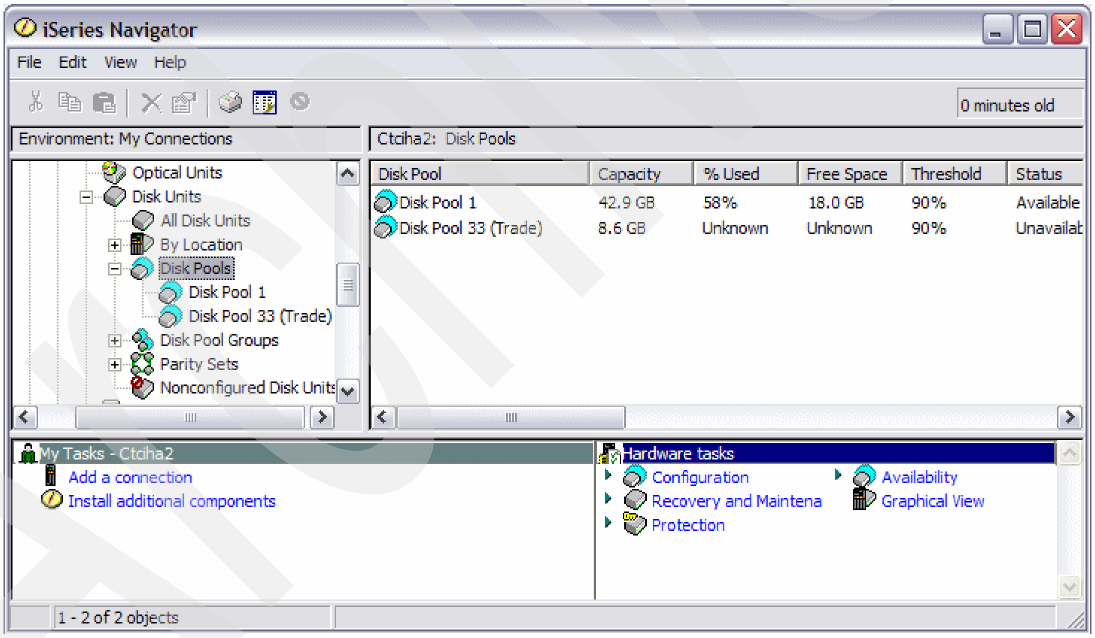


Warning Message: Warning - system is not in a device domain group. (Error Code: 507)

To view more information about a message, select the message and click the Details button. To Continue with this action, click Continue. To cancel this action, click Cancel

Figure 5-15 Warning message after pool creation

You should now see the new disk pool under the Disk Pools option in iSeries Navigator as shown in Figure 5-16.



Disk Pool	Capacity	% Used	Free Space	Threshold	Status
Disk Pool 1	42.9 GB	58%	18.0 GB	90%	Available
Disk Pool 33 (Trade)	8.6 GB	Unknown	Unknown	90%	Unavailab

1 - 2 of 2 objects

Figure 5-16 Disk pool creation completed

With the disk pool creation now finished, we have the IASP with the name Trade. Our next step is to configure the cluster.

5.3.2 Restoring the Trade database

At this time, you can create or restore the application database. In our case, we restore the Trade database. See Appendix F, “Additional material” on page 297, for more information about restoring the Trade database.

5.3.3 Creating a cluster

A cluster of two or more system greatly increases your overall system availability. In our scenario, the iSeries cluster is created to support the data availability. A cluster describes the nodes that will participate in the clusters (machines) and the respective interface address used for cluster resource communication. The cluster interface address is an IP address that is used by Cluster Resource Services to communicate with other nodes in the cluster.

To prepare for configuring clusters with iSeries Navigator, you must ensure that certain conditions exist in the iSeries setup and configuration. Prior to creating a cluster, make sure that the TCP configuration is correct on the system.

TCP/IP requirements

Before creating a cluster, check the following TCP/IP configuration parameters on your iSeries systems:

1. Start TCP/IP on every node that you plan to include in the cluster using the Start TCP/IP (`STRTCP`) command.
2. Configure the TCP loopback address (127.0.0.1). Verify that it shows a status of *Active*. Using the Work with TCP/IP Network Status (`WRKTCPSTS`) command, verify this status on every node in the cluster.
3. Verify that the IP addresses used for clustering to a given node show a status of *Active*, by using the `WRKTCPSTS` command on the subject node.

> **Optional step:** Geographic mirroring uses several communications transports. If you want to configure fault tolerance for network routes used by geographic mirroring, you need to perform several additional steps, as described in “Configuring fault tolerance for network routes using virtual IP and proxy ARP” on page 288 and “Associating the local IP interfaces with the virtual IP interface” on page 290.

4. Verify that INETD is active on all nodes in the cluster, by using the command:

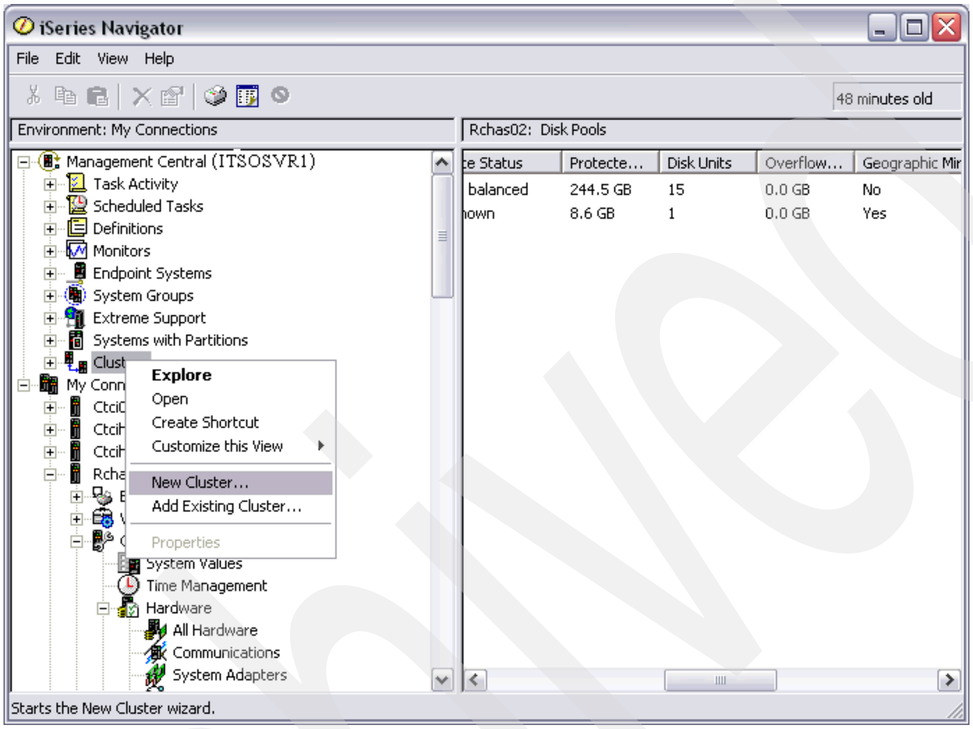
   ```
   STRTCPSVR *INETD
   ```

   You can verify this by the presence of a QTOGINTD (User QTCP) job in the Active Jobs list on the subject node.
5. Verify that the local and any remote nodes are reachable. Run the PING utility, using the IP addresses for clustering, to ensure that network routing is active.
6. Verify that ports 5550 and 5551 are not being used by other applications. These ports are reserved for IBM clustering. View port usage by using the `WRKTCPSTS` command. Port 5550 is opened by clustering and is in a *Listen* state after INETD is started.

Performing the steps to create a cluster

Now that we have verified that the TCP/IP configuration is correct and have an IASP, let's create a cluster from the Management Central GUI.

1. As shown in Figure 5-17, in iSeries Navigator, expand **Management Central**. Right-click **Cluster** and select **New Cluster**.

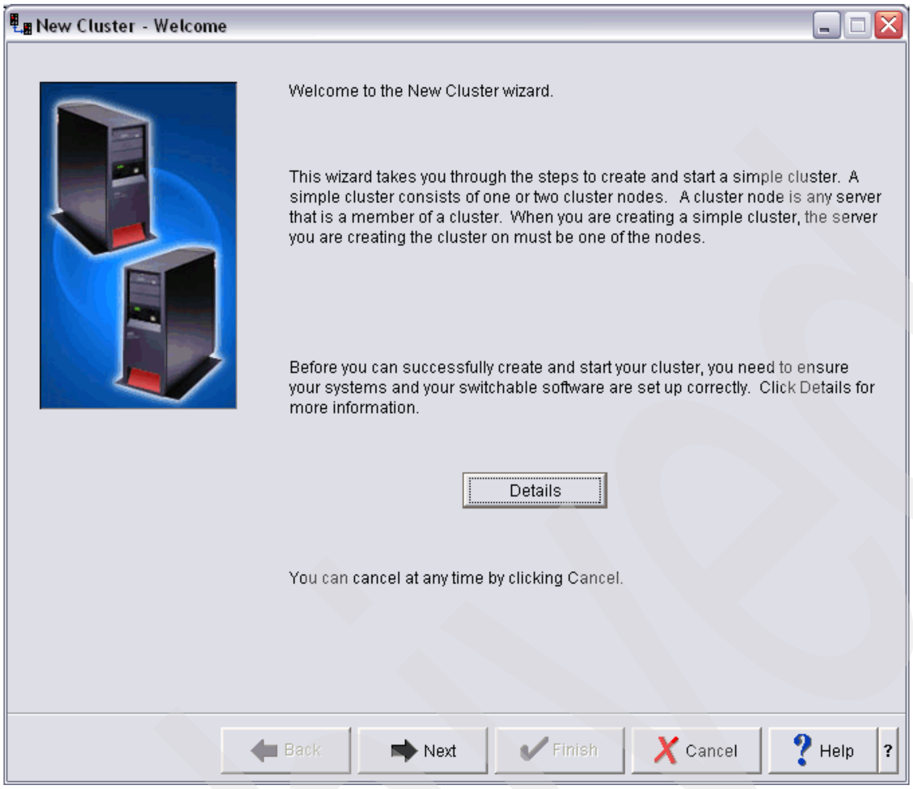


e Status	Protecte...	Disk Units	Overflow...	Geographic Mir
balanced	244.5 GB	15	0.0 GB	No
own	8.6 GB	1	0.0 GB	Yes

Starts the New Cluster wizard.

Figure 5-17 Creating a cluster

2. In the New Cluster wizard Welcome panel (Figure 5-18), click **Next**.



Welcome to the New Cluster wizard.

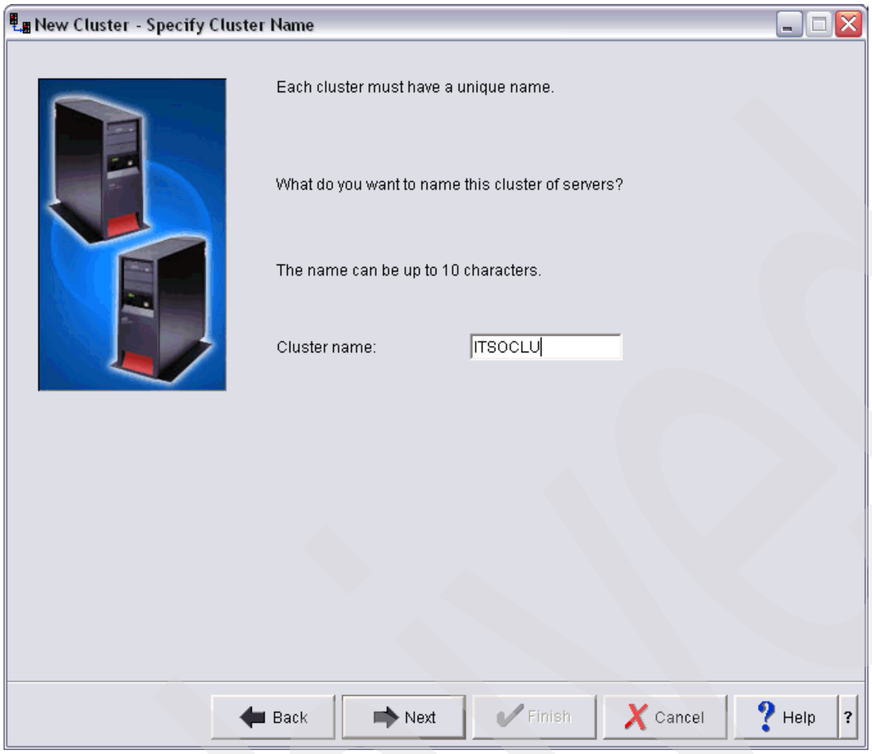
This wizard takes you through the steps to create and start a simple cluster. A simple cluster consists of one or two cluster nodes. A cluster node is any server that is a member of a cluster. When you are creating a simple cluster, the server you are creating the cluster on must be one of the nodes.

Before you can successfully create and start your cluster, you need to ensure your systems and your switchable software are set up correctly. Click Details for more information.

You can cancel at any time by clicking Cancel.

Figure 5-18 Welcome panel

3. In the Specify Cluster Name panel (Figure 5-19), type a name for the cluster and click **Next**.



Each cluster must have a unique name.

What do you want to name this cluster of servers?

The name can be up to 10 characters.

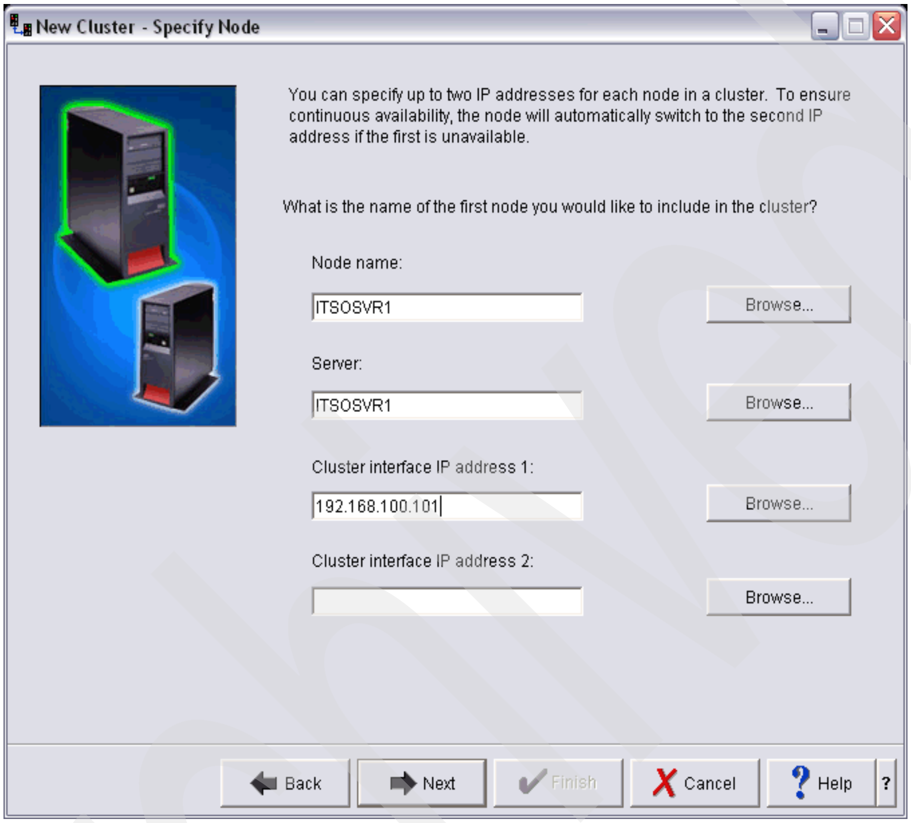
Cluster name: ITSOCLU

Figure 5-19 Specifying the cluster name

4. In the Specify Node panel (Figure 5-20), specify the first node to add to the cluster. We add ITSOSVR1 as the first node by providing the following information:

   – Node name and Server: `ITSOSVR1`
   – IP address: `192.168.100.101`

   Click **Next**.



New Cluster - Specify Node

You can specify up to two IP addresses for each node in a cluster. To ensure continuous availability, the node will automatically switch to the second IP address if the first is unavailable.

What is the name of the first node you would like to include in the cluster?

Node name: ITSOSVR1

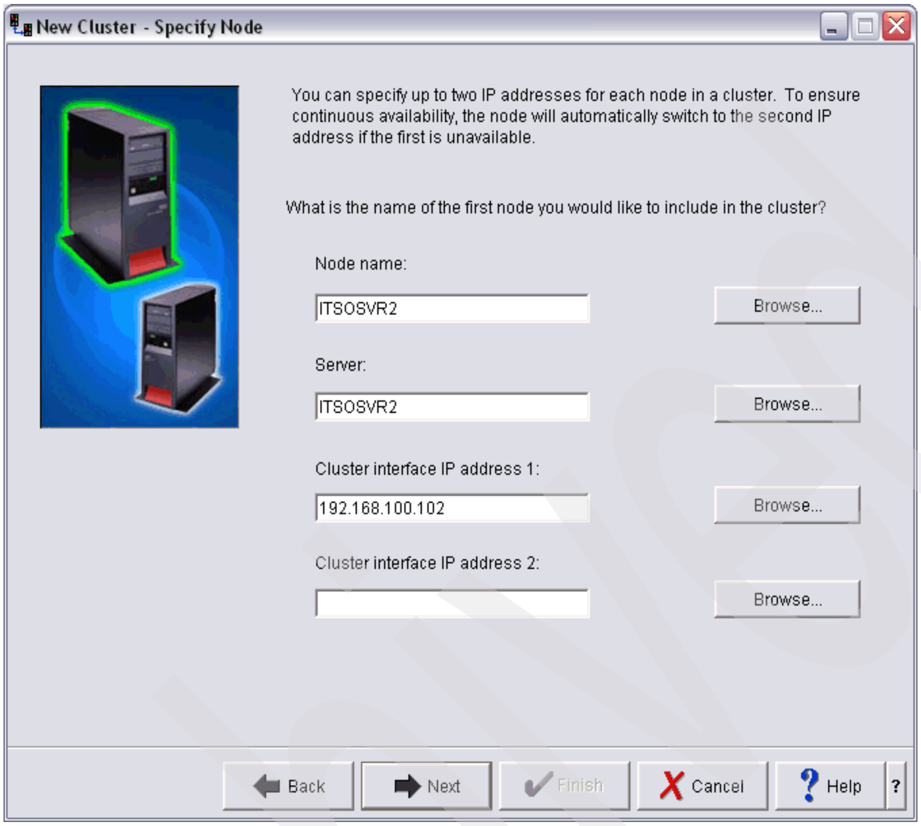
Server: ITSOSVR1

Cluster interface IP address 1: 192.168.100.101

Cluster interface IP address 2:

Figure 5-20 Specifying the ITSOSVR1 node in the cluster

5. In the next panel (Figure 5-21), specify the second, backup node, ITSCOSVR2, and the IP interface for node communication, which in this scenario is `192.168.100.102`. Click **Next**.



New Cluster - Specify Node

You can specify up to two IP addresses for each node in a cluster. To ensure continuous availability, the node will automatically switch to the second IP address if the first is unavailable.

What is the name of the first node you would like to include in the cluster?

Node name: ITSOSVR2

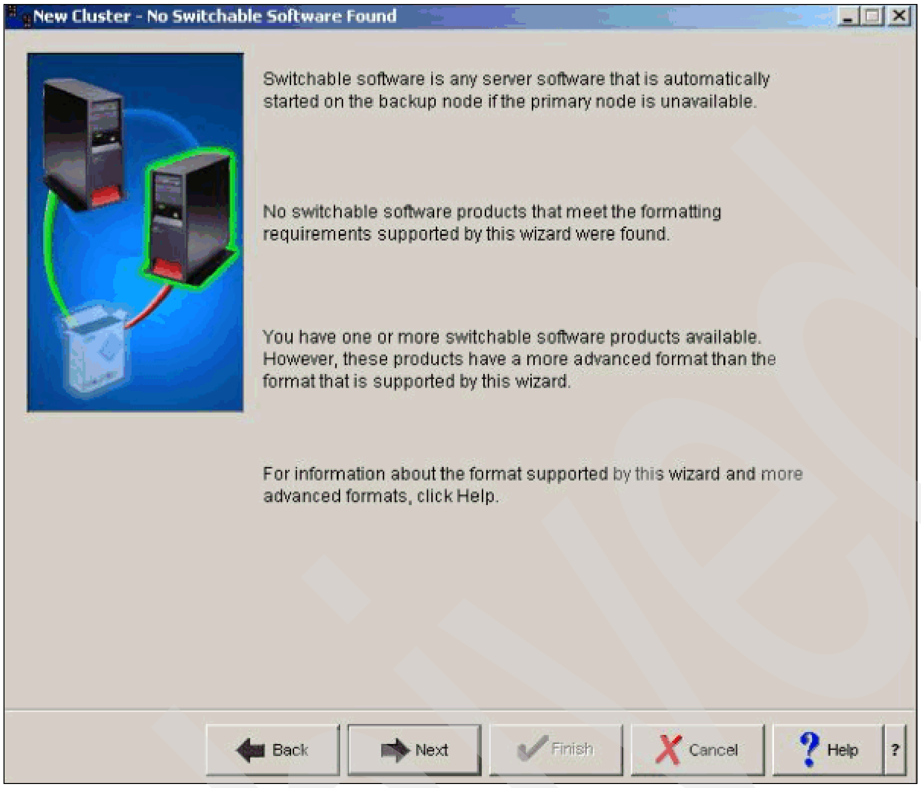
Server: ITSOSVR2

Cluster interface IP address 1: 192.168.100.102

Cluster interface IP address 2:

Figure 5-21 Specifying the backup node

6. In the No Switchable Software Found panel (Figure 5-22), click **Next** to create the cluster.



Switchable software is any server software that is automatically started on the backup node if the primary node is unavailable.

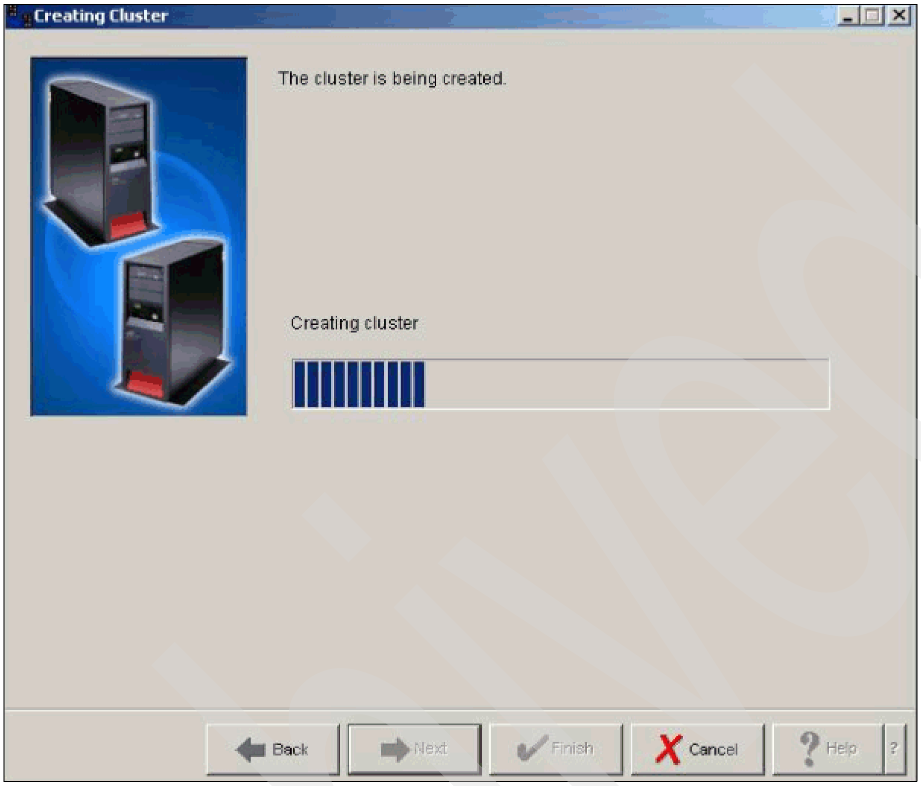
No switchable software products that meet the formatting requirements supported by this wizard were found.

You have one or more switchable software products available. However, these products have a more advanced format than the format that is supported by this wizard.

For information about the format supported by this wizard and more advanced formats, click Help.

Figure 5-22 No Switchable Software Found panel

7. The Creating Cluster panel (Figure 5-23) opens next, showing the cluster creation status. After the process completes, click **Next**. You should see that the cluster has been created successfully.

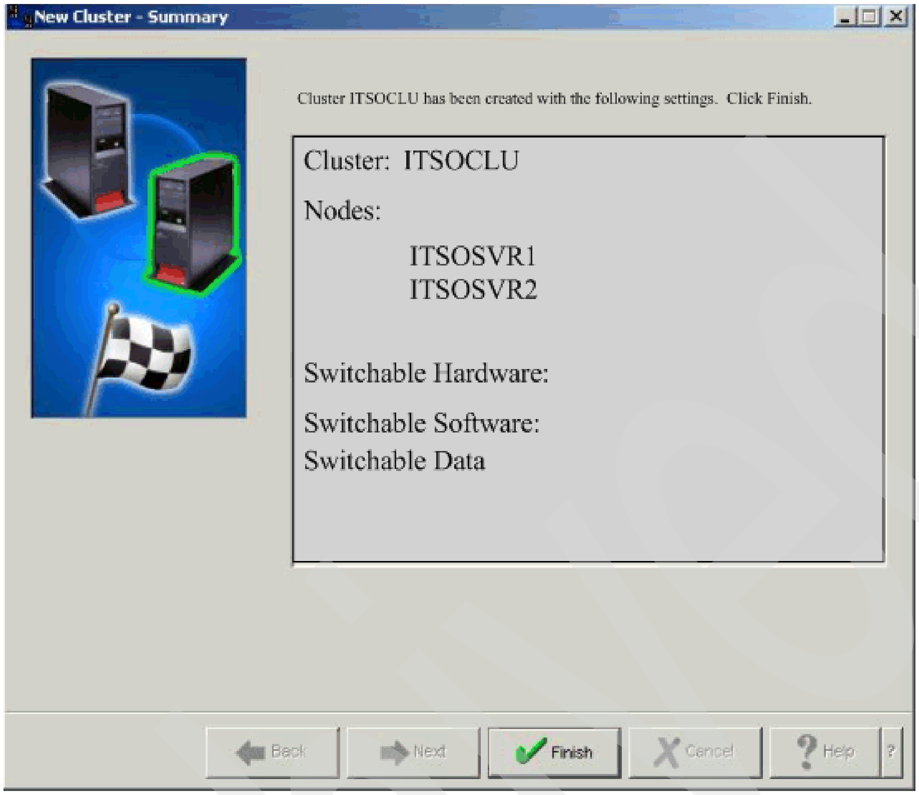


The cluster is being created.

Creating cluster

Figure 5-23 Cluster creation in progress

8. In the Summary panel (Figure 5-24), review the information and click **Finish**.



Cluster ITSOCLU has been created with the following settings. Click Finish.

Cluster: ITSOCLU

Nodes:

ITSOSVR1
ITSOSVR2

Switchable Hardware:

Switchable Software:

Switchable Data

Figure 5-24 Confirmation of creating the cluster ITSOCLU

Now that we have created the cluster, the current XSM configuration resembles the diagram shown in Figure 5-25.

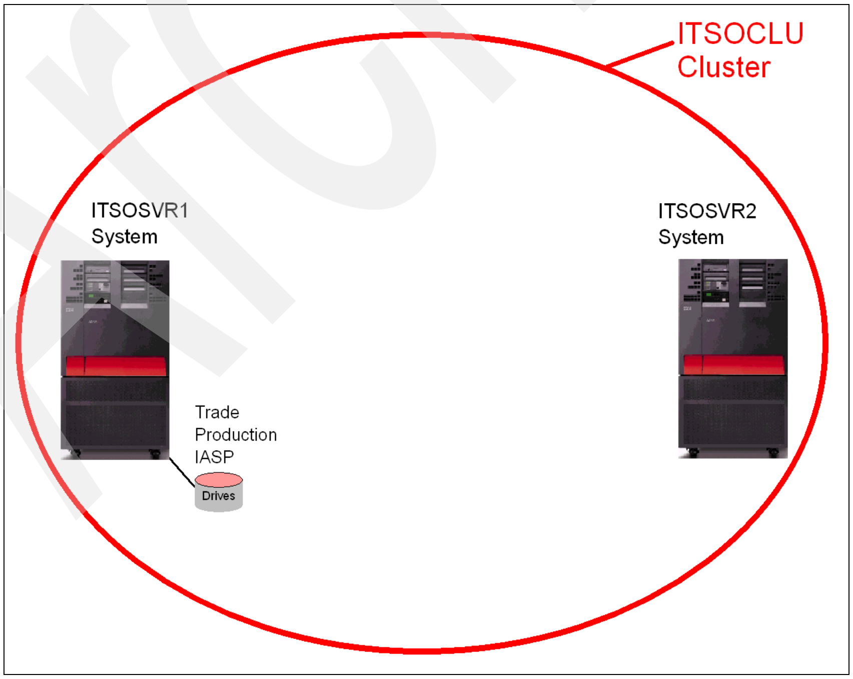


Figure 5-25 Current cluster configuration

5.3.4 Setting up a CRG

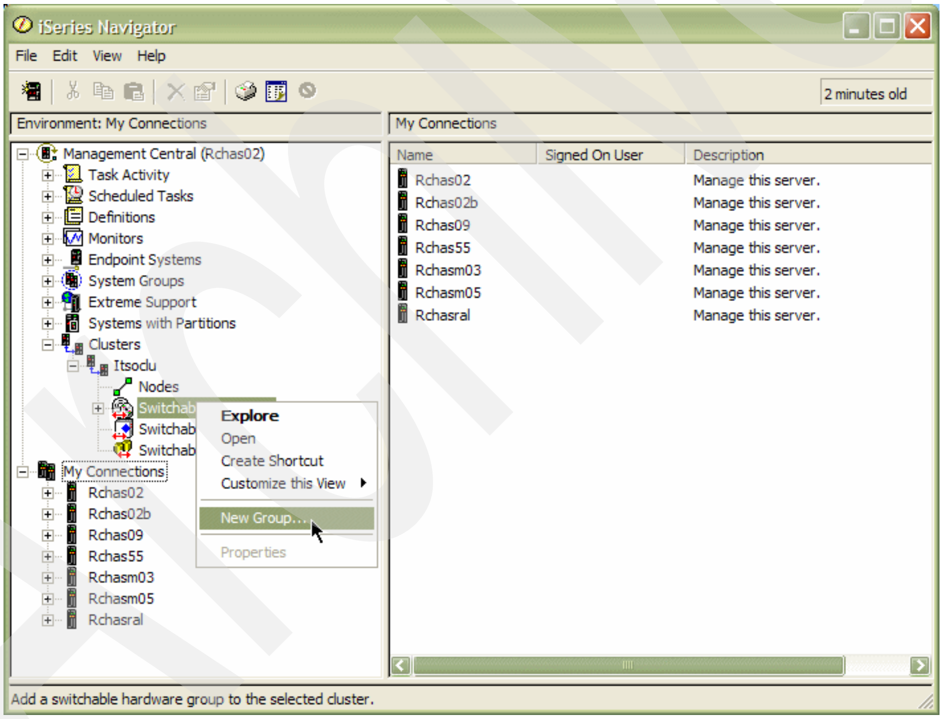
A CRG, also known as a switchable hardware group, is an OS/400 system object that represents a set or grouping of cluster resources that define actions to be taken during a switchover or failover for this group of the cluster devices. The group identifies two important elements:

- Recovery domain
- CRG exit program

  This element manages cluster-related events for that group. One such event is moving an access point from one node to another node.

CRG objects are defined as data resilient, application resilient, or device resilient. *Data resiliency* enables multiple copies of data to be maintained on more than one node in a cluster. It also enables the point of access to be changed to a backup node. *Application resiliency* enables an application program to be restarted on either the same node or a different node in the cluster. *Device resiliency* enables a device resource to be moved (switched) to a backup node. In our example, we only need to create a device CRG since we are using a switchable disk pool in our configuration.

1. As shown in Figure 5-26, expand **Management Central** → **Clusters** → ***your cluster***. Right-click **Switchable Hardware** and select **New Group**.

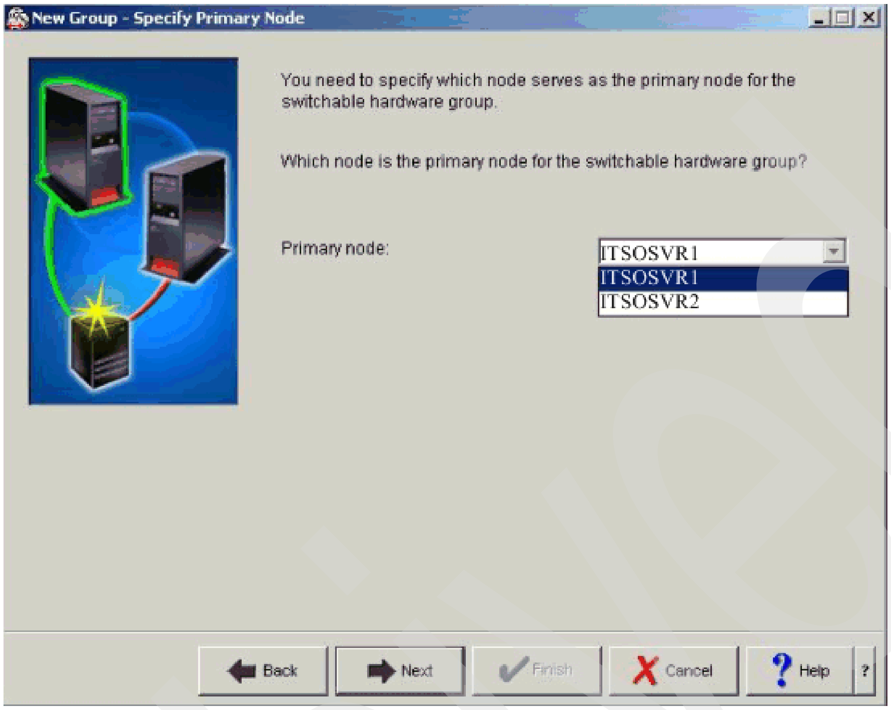


Name	Signed On User	Description
Rchas02		Manage this server.
Rchas02b		Manage this server.
Rchas09		Manage this server.
Rchas55		Manage this server.
Rchasm03		Manage this server.
Rchasm05		Manage this server.
Rchasral		Manage this server.

Add a switchable hardware group to the selected cluster.

Figure 5-26 Creating the CRG

2. In the Specify Primary Node panel (Figure 5-27), select the primary node in the CRG and click **Next**.



New Group - Specify Primary Node

You need to specify which node serves as the primary node for the switchable hardware group.

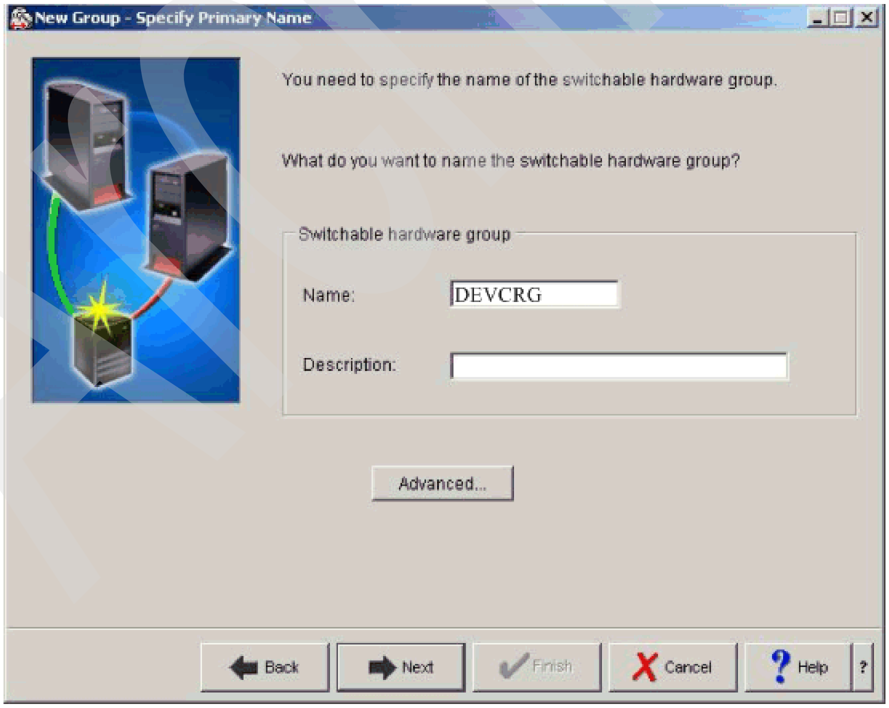
Which node is the primary node for the switchable hardware group?

Primary node: ITSOSVR1

ITSOSVR1
ITSOSVR2

Figure 5-27 Setting the primary node in CRG

3. In the Specify Primary Name panel (Figure 5-28), specify the CRG name and click **Next**.



New Group - Specify Primary Name

You need to specify the name of the switchable hardware group.

What do you want to name the switchable hardware group?

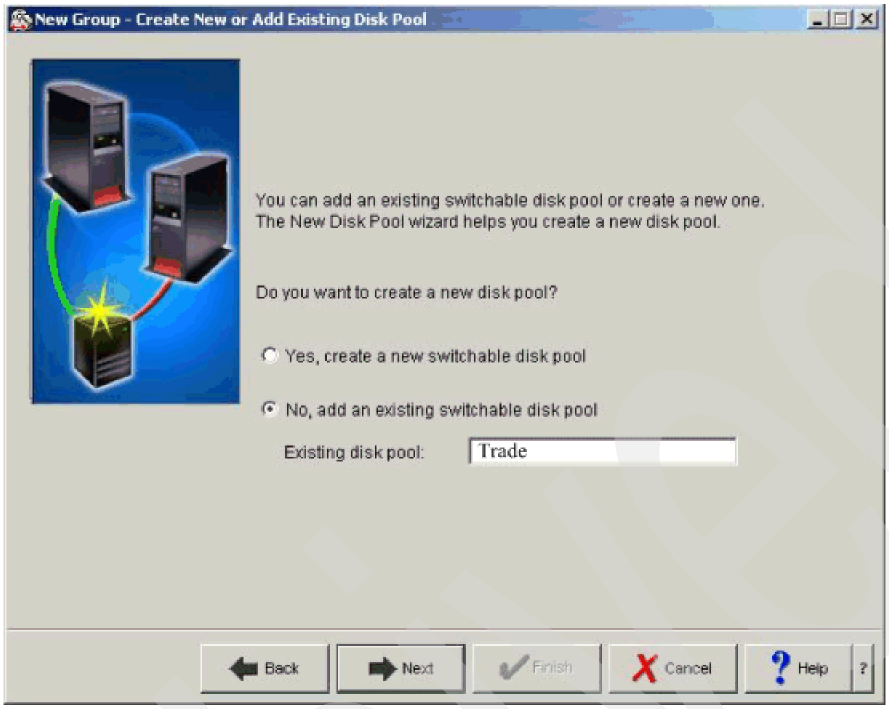
Switchable hardware group

Name: DEVCRG

Description:

Figure 5-28 Specifying the CRG name

4. In the Create New or Add Existing Disk Pool panel (Figure 5-29), specify the existing disk pool to add to CRG and click **Next**.



You can add an existing switchable disk pool or create a new one. The New Disk Pool wizard helps you create a new disk pool.

Do you want to create a new disk pool?

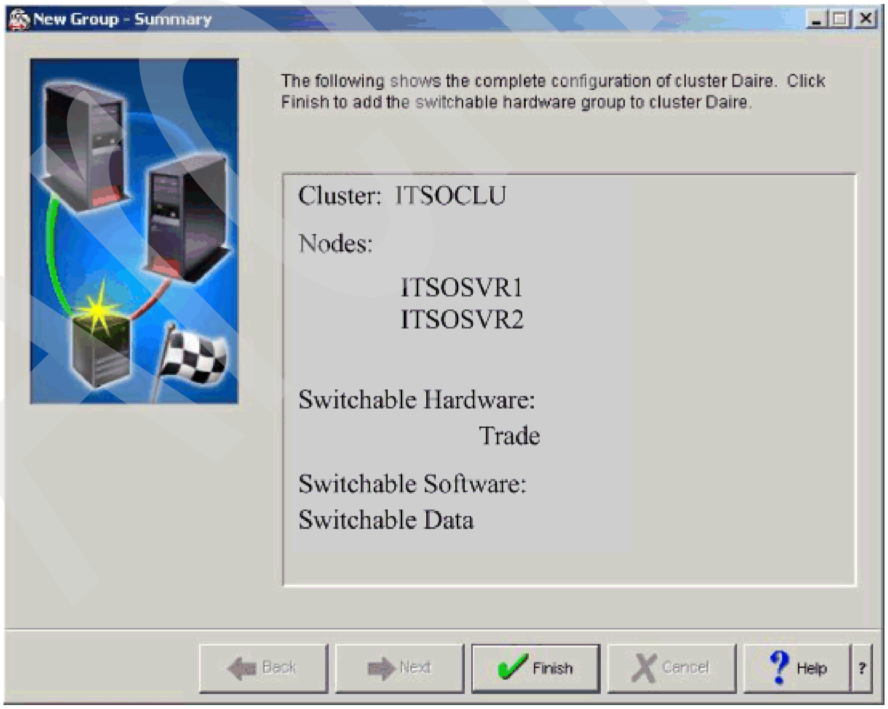
○ Yes, create a new switchable disk pool

● No, add an existing switchable disk pool

Existing disk pool: Trade

Figure 5-29 Adding an existing Trade disk pool to CRG

5. On the Summary panel (Figure 5-30), confirm the creation of the CRG and click **Finish**.



The following shows the complete configuration of cluster Daire. Click Finish to add the switchable hardware group to cluster Daire.

Cluster: ITSOCLU

Nodes:

ITSOSVR1
ITSOSVR2

Switchable Hardware:
Trade

Switchable Software:
Switchable Data

Figure 5-30 Confirming the creation of the CRG

Now that we have created the CRG, our configuration resembles the diagram in Figure 5-31.

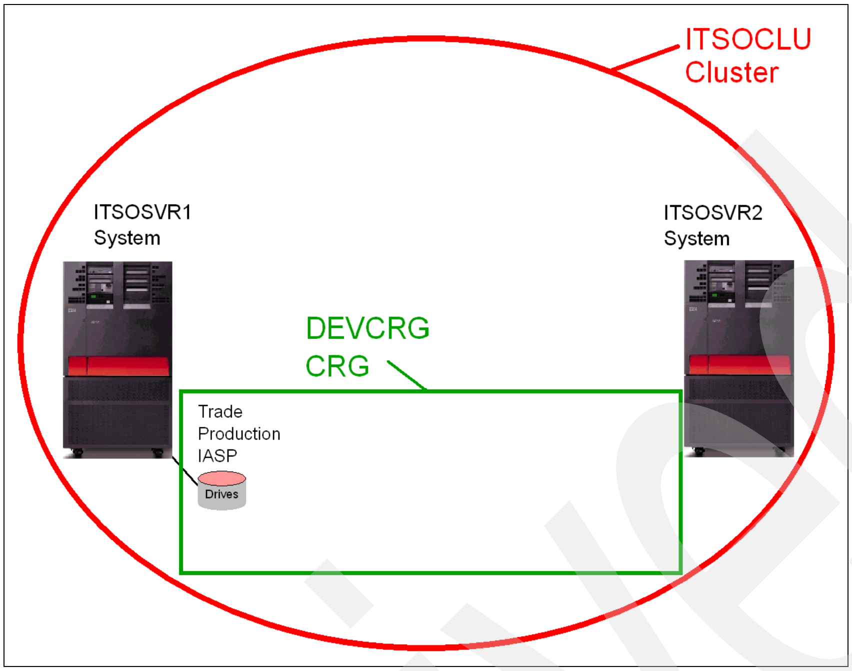


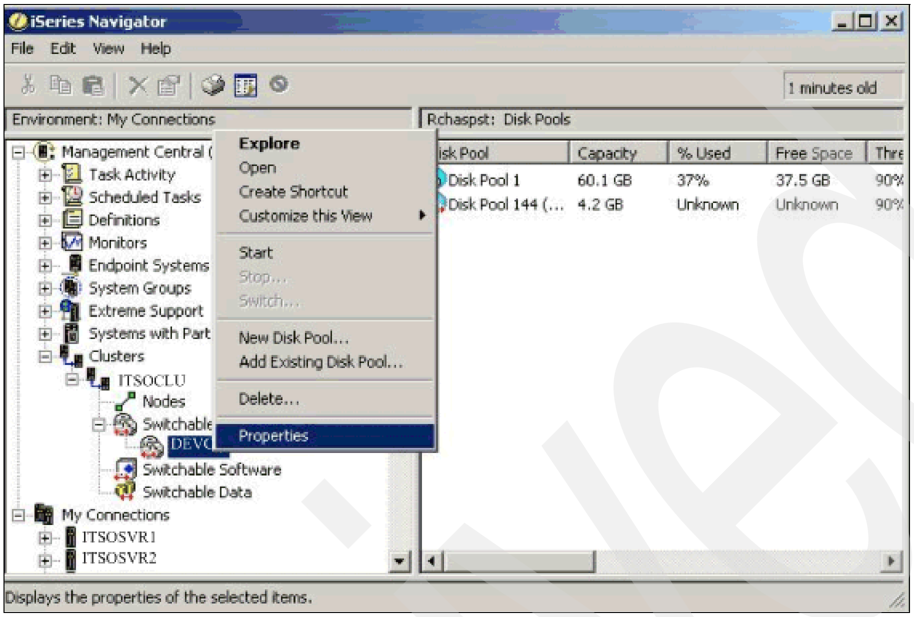
Figure 5-31 Current XSM configuration after creating the CRG

5.3.5 Defining a recovery domain for CRG

A recovery domain is a subset of cluster nodes that are grouped together in a CRG for a common purpose such as performing a recovery action. A domain represents those nodes of the cluster from which cluster resource can be accessed. This subset of cluster nodes that is assigned to a particular CRG either supports the primary point of access, secondary (backup) point of access, or replicate.

Perform the following steps to configure the recovery domain:

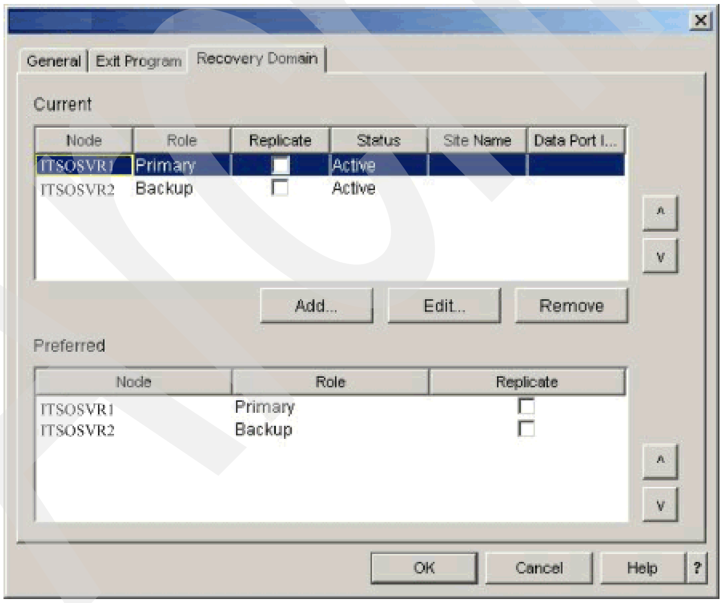
1. As shown in Figure 5-32, expand **Management Central** → **Clusters** → ***your cluster*** → **Switchable Hardware**. Right-click ***your CRG*** and select **Properties**.



isk Pool	Capacity	% Used	Free Space	Thre
Disk Pool 1	60.1 GB	37%	37.5 GB	90%
Disk Pool 144 (...	4.2 GB	Unknown	Unknown	90%

Figure 5-32 Initiating the configuration of the recovery domain for CRG

2. In the next window (Figure 5-33), click the **Recovery Domain** tab. On this tab, you specify the primary node and backup node roles.



Current

Node	Role	Replicate	Status	Site Name	Data Port I...
ITSOSVR1	Primary	□	Active		
ITSOSVR2	Backup	□	Active		

Preferred

Node	Role	Replicate
ITSOSVR1	Primary	□
ITSOSVR2	Backup	□

Figure 5-33 Specifying the primary and backup nodes in CRG

a. Add the node site name and data port address for the primary node.

> **Optional step:** Geographic mirroring uses several communications transports. If you want to configure fault tolerance for network routes used by geographic mirroring, you must perform an additional step, which is explained in “Updating your cluster configuration” on page 291. Complete this step for the primary and backup nodes. It is a consequence of the optional step explained on page 109.

i. Select the first node, which is **ITSOSVR1** in this example, and click **Edit**.

ii. The General window (Figure 5-34) opens. In the Site name field, type the site name for the node (1).

The site contains a subset of recovery domain nodes in the physical location. All nodes at a site have access to the same copy of the ASP. For this example, we use `SITE1` for the primary node site name.

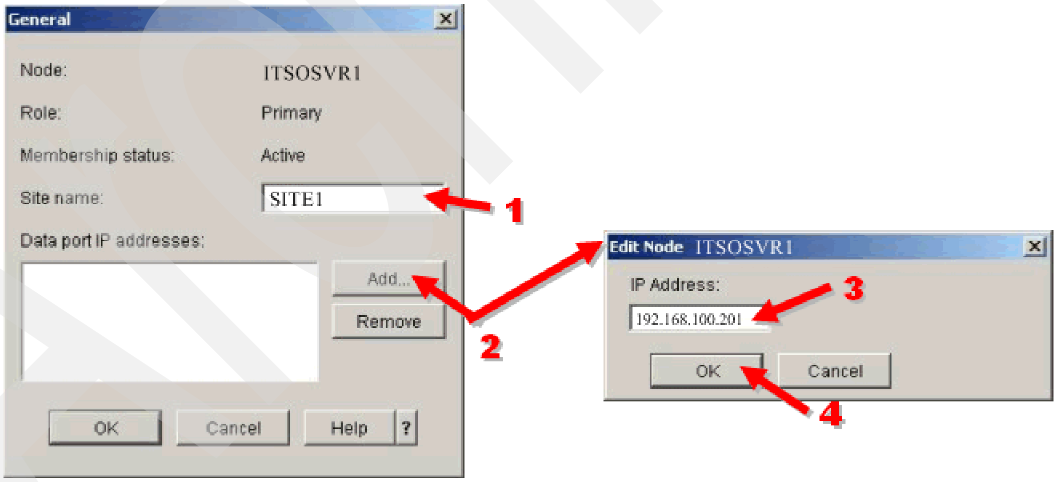
Click **Add** to add the data port (2).

iii. The Edit Node window opens. In the IP Address field, type the address of the dataport (3).

The data port IP address is used to send updates from a source node that has the production copy of ASP to a target node that owns the mirror copy of the ASP. In this case, the IP address that is used for the primary node is `192.168.100.201`. Make sure that this is a dedicated communication line for the replication. XSM can support up to four communication lines; we recommend that you configure at least two lines for high-availability reasons.

Click **OK** (4).

iv. Back on the General window, click **OK**.



General

Node: ITSOSVR1

Role: Primary

Membership status: Active

Site name: SITE1

Data port IP addresses:

Edit Node ITSOSVR1

IP Address: 192.168.100.201

Figure 5-34 Adding a site name and data port for the primary node in CRG

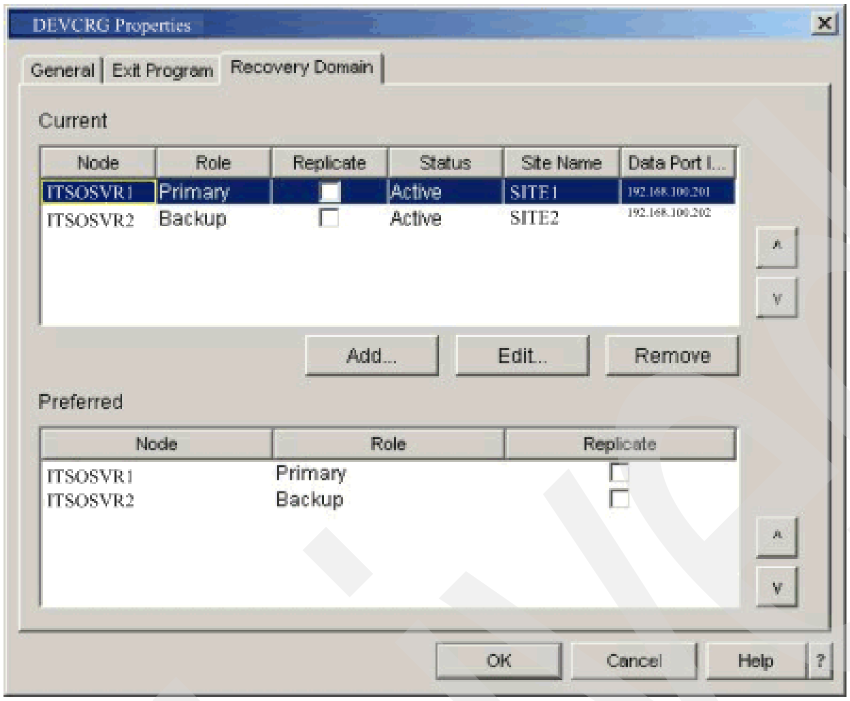
b. Add the site name and data port properties for backup node in CRG. Edit the properties for the ITSOSVR2 backup node, and specify the following attributes:

i. In the General window, for Site name, type `SITE2`. Click **Add**.

ii. In the Edit Node window, type the IP address `192.168.100.202` for the data port. Click **OK**.

iii. Back in the General window, click **OK**.

After qualifying the site names and data ports for the primary and backup node, you see the changes in the DEVCRG Properties window (Figure 5-35).



Current

Node	Role	Replicate	Status	Site Name	Data Port I...
ITSOSVR1	Primary	□	Active	SITE1	192.168.100.201
ITSOSVR2	Backup	□	Active	SITE2	192.168.100.202

Preferred

Node	Role	Replicate
ITSOSVR1	Primary	□
ITSOSVR2	Backup	□

Figure 5-35 Specifying the primary and backup roles with site names

After creating the recovery domain, the configuration now resembles the diagram in Figure 5-36.

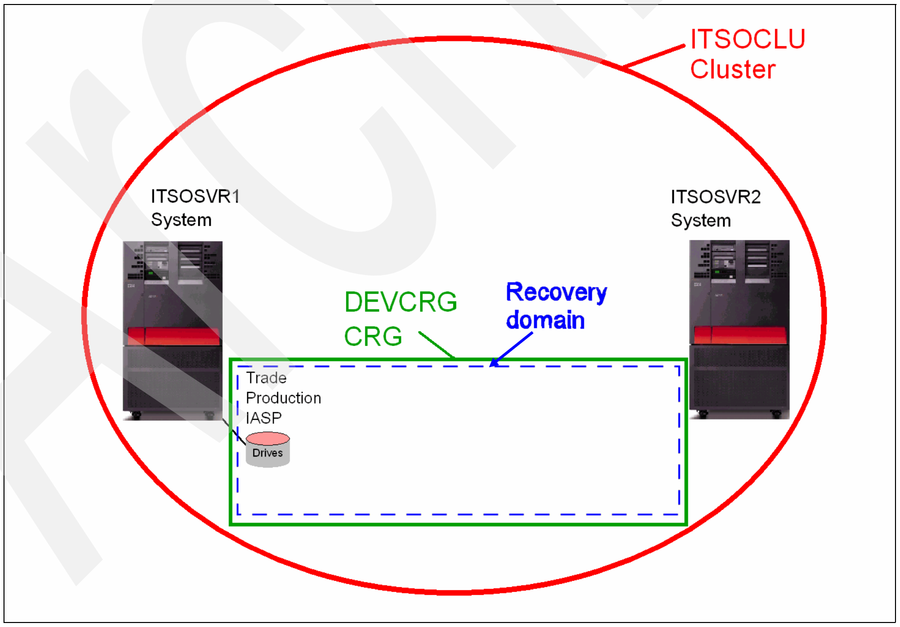


Figure 5-36 Configuration after creating a recovery domain

Setting the takeover IP address

The idea of a cluster is based on the fact that multiple resource in the cluster are accessed through the same IP address. In case of a failover, the cluster software is responsible for bringing down the failed resource and activating a backup resource. During this process, an IP address that is associated with the failed resource is reassigned to a backup resource. Such an IP address that can be associated with multiple resource in the iSeries cluster is called a *takeover IP address*.

In our example, we have to define a takeover IP address for our two systems in the cluster. Perform the following steps to configure a takeover IP address.

1. As shown in Figure 5-37, follow these steps:

   a. Expand **Management Central** → **Clusters** → ***your cluster*** → **Switchable Hardware**. Then click ***your CRG device***.

   b. In the right pane, right-click ***your IASP*** (Trade in our example) and select **Properties**.

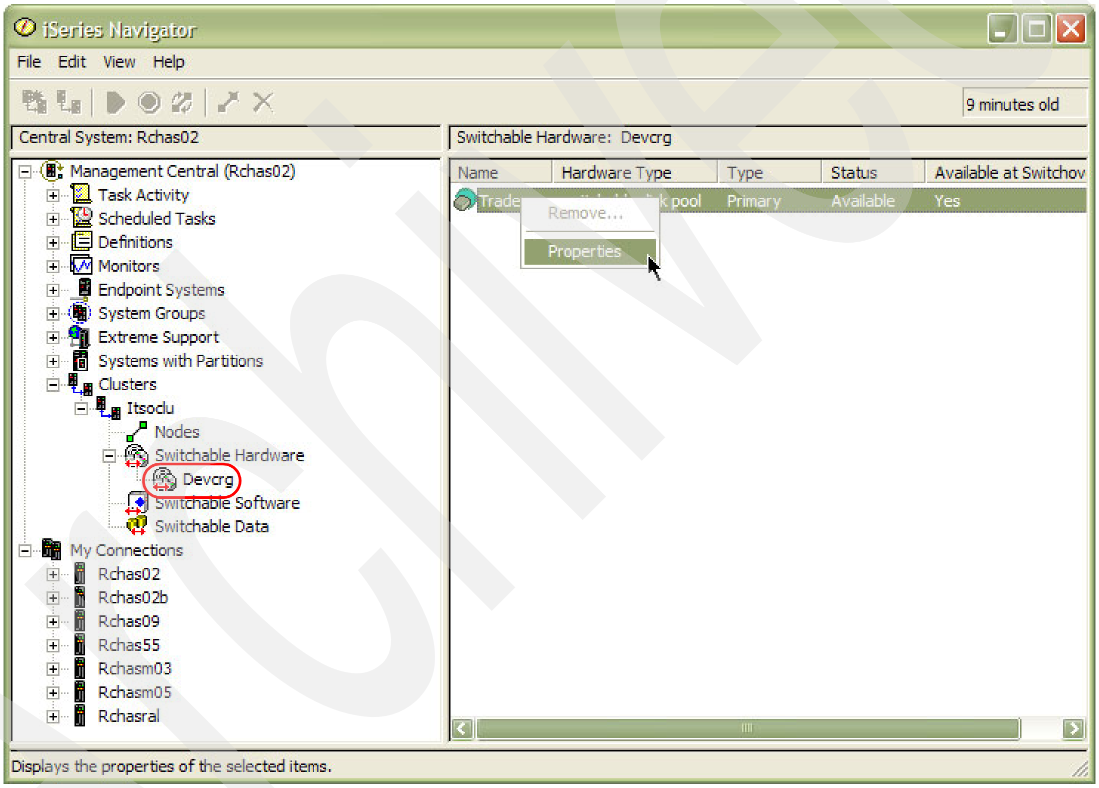
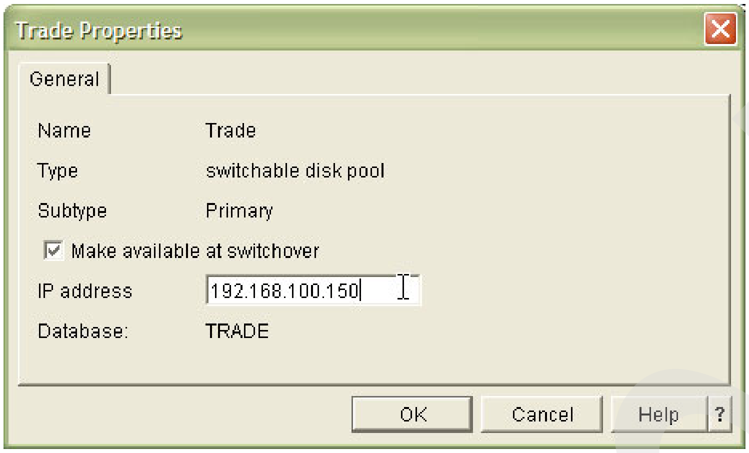


Figure 5-37 Opening the Properties window

2. In the Properties window (Figure 5-38), type the IP address that will define the takeover IP address for your cluster. Click **OK**.



Name Trade

Type switchable disk pool

Subtype Primary

☑ Make available at switchover

IP address 192.168.100.150

Database: TRADE

Figure 5-38 Setting the takeover IP

From this point forward, when you need to access data located on this cluster, you must use the takeover IP address. We demonstrate how to do that when we configure the data source in WebSphere Application Server.

5.3.6 Configuring geographic mirroring

Geographic mirroring is a function that generates a mirror copy of an independent disk pool on a system that is (optionally) geographically distant from the originating site for availability or protection purposes. The nodes participating in geographic mirroring must be part of a cluster. When geographic mirroring is configured, the mirror copy has the same disk pool number and name as the original disk pool, the production copy. The two disk pools must have similar disk capacities (a difference of less than 5%), but the mirror copy may have different numbers and types of disk units, as well as different types of disk protection.

When geographic mirroring is active, changes to the production copy data are transmitted to the mirror copy across TCP/IP connections. Changes can be transmitted either synchronously or asynchronously. If the user chooses synchronous mode, the client waits until the operation is complete on both the source and target systems. If the user chooses asynchronous mode, the client must wait only until the operation is received for processing on the target system. The synchronization with the mirror copy is done asynchronously.

To maintain the data integrity of the mirror copy, the user cannot access the mirror copy while geographic mirroring is being performed. The user can detach the mirror copy to perform saves, reports, and data mining. However, the mirror copy must be resynchronized with the production copy after it is reattached. Synchronization can be a lengthy process if you have a large data set in the disk pool or relatively slow (such as T1 speed) communications lines are used for the XSM replication processing between cluster nodes.

When you configure the cluster for geographic mirroring, you have many options for defining the availability and protection of the independent disk pool. When you create the switchable hardware group, you list the order of the backup systems to which the independent disk pool will failover or switchover. If the primary node switches to a backup node at the remote site, the mirror copy on the backup node changes roles to become the production copy. The production copy is now accessible for updates on the remote system. If the independent disk pool is part of a disk pool group, all of the disk pools in the group will switch over together.

Geographic mirroring is a subfunction of XSM, which is part of OS/400 (5722-SS1 Option 41, High Available Switchable Resources).

Follow these steps to configure geographic mirroring.

1. Initiate the geographic mirroring configuration from iSeries Navigator as shown in Figure 5-39.

   a. Expand **My Connections** → ***your system*** → **Configuration and Service** → **Hardware** → **Disk Units** → **Disk Pools**.

   b. Use the SST user ID and password to access the disk configuration.

   c. Right-click the disk pool that was created earlier on primary node ITSOSVR1 and select **Geographic Mirroring** → **Configure Geographic Mirroring**.

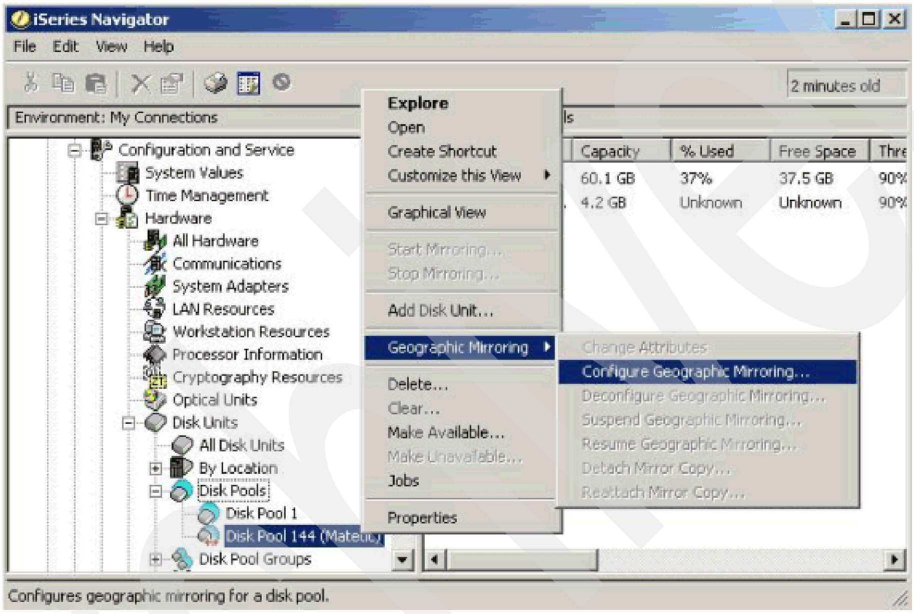
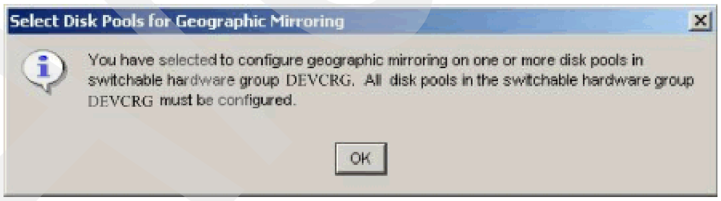


Figure 5-39 initiating geographic mirroring via iSeries Navigator

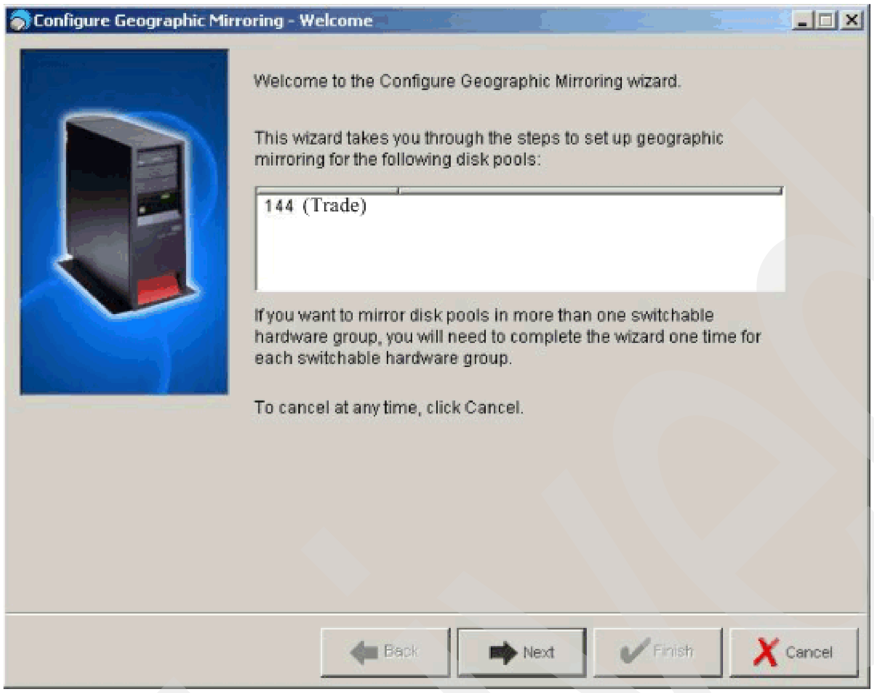
2. In the Select Disk Pools for Geographic Mirroring window (Figure 5-40), click **OK**.


Select Disk Pools for Geographic Mirroring

You have selected to configure geographic mirroring on one or more disk pools in switchable hardware group DEVCRG. All disk pools in the switchable hardware group DEVCRG must be configured.

Figure 5-40 Confirming the geographic configuration

3. The Configure Geographic Mirroring wizard Welcome panel (Figure 5-41) opens. It shows the disk pool and IASP name that were defined previously. Click **Next**.



Welcome to the Configure Geographic Mirroring wizard.

This wizard takes you through the steps to set up geographic mirroring for the following disk pools:

144 (Trade)

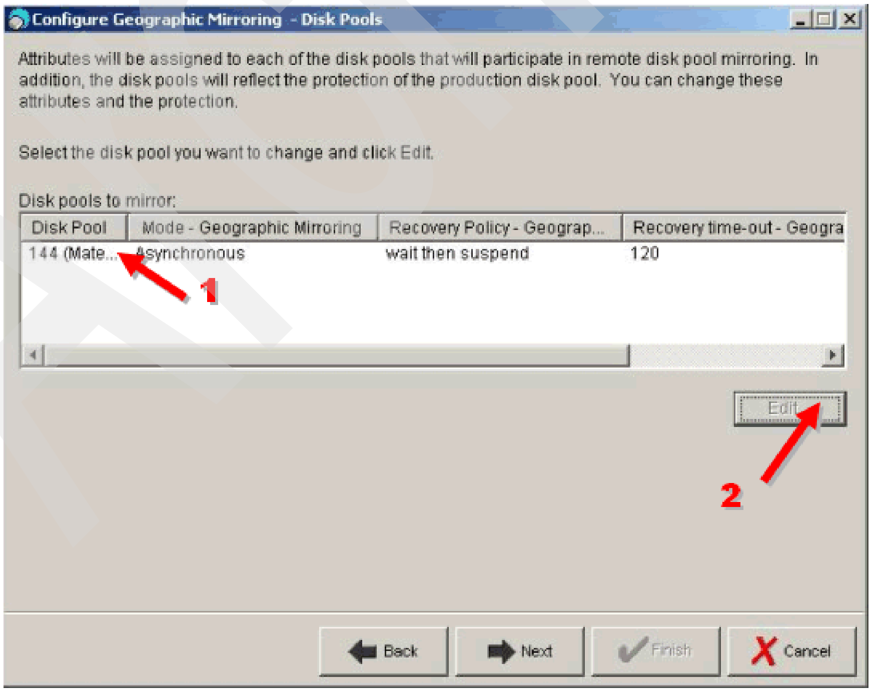
If you want to mirror disk pools in more than one switchable hardware group, you will need to complete the wizard one time for each switchable hardware group.

To cancel at any time, click Cancel.

Figure 5-41 Configure Geographic Mirroring Welcome panel

4. In the Disk Pools panel (Figure 5-42), change some of the attributes of the currently configured disk pool.

   a. Select the defined disk pool (**1**) and click **Edit** (**2**).



Attributes will be assigned to each of the disk pools that will participate in remote disk pool mirroring. In addition, the disk pools will reflect the protection of the production disk pool. You can change these attributes and the protection.

Select the disk pool you want to change and click Edit.

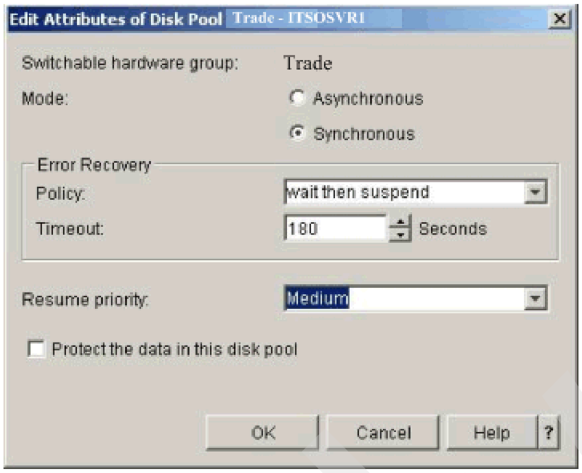
Disk pools to mirror:

Disk Pool	Mode - Geographic Mirroring	Recovery Policy - Geograp...	Recovery time-out - Geogra
144 (Mate...	Asynchronous	wait then suspend	120

Figure 5-42 Changing the disk pool attributes

b. In the Edit Attributes of Disk Pool window (Figure 5-43), modify the attributes for the Trade disk pool.

   i. For Mode, select **Synchronous**.
   ii. For Error Recovery Time-out, select **180**.
   iii. Click **OK** to confirm the attribute changes.



Switchable hardware group: Trade

Mode: ○ Asynchronous ● Synchronous

Error Recovery

Policy: wait then suspend

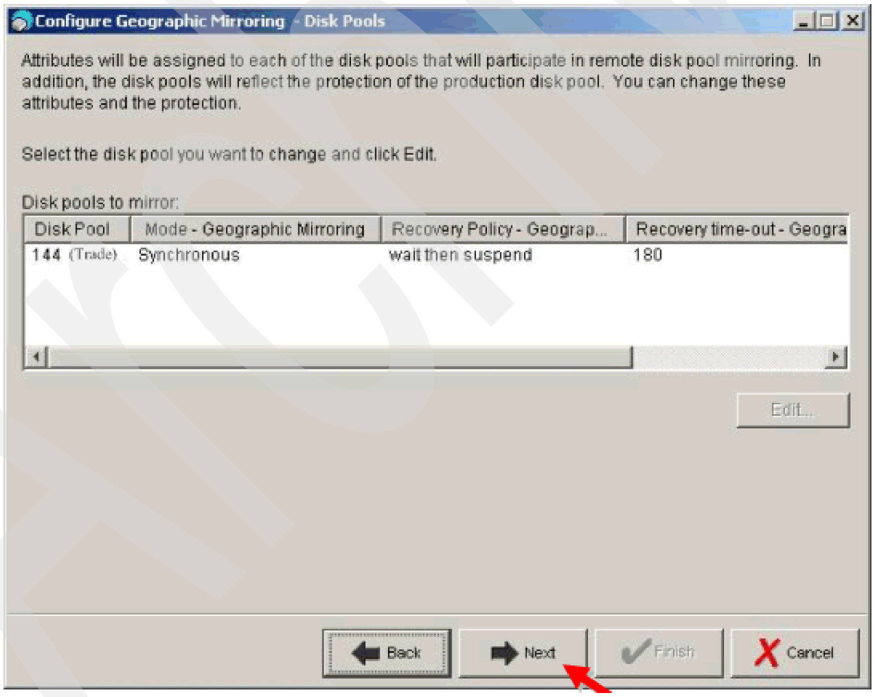
Timeout: 180 Seconds

Resume priority: Medium

□ Protect the data in this disk pool

Figure 5-43 Modifying the attributes of the Trade disk pool

c. Back on the Disk Pools panel (Figure 5-44), confirm the changes and then click **Next**.



Attributes will be assigned to each of the disk pools that will participate in remote disk pool mirroring. In addition, the disk pools will reflect the protection of the production disk pool. You can change these attributes and the protection.

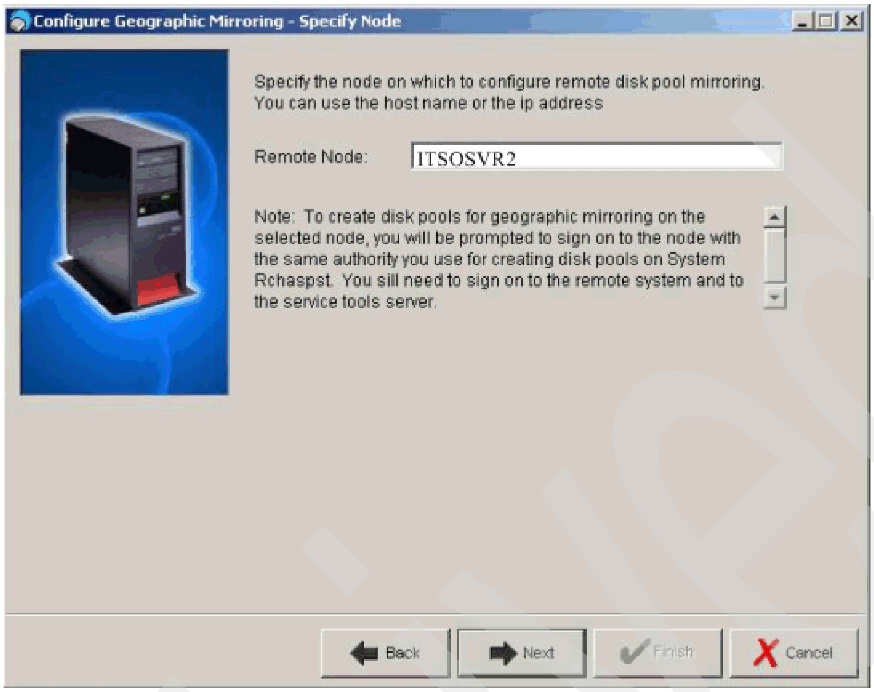
Select the disk pool you want to change and click Edit.

Disk pools to mirror:

Disk Pool	Mode - Geographic Mirroring	Recovery Policy - Geograp...	Recovery time-out - Geogra
144 (Trade)	Synchronous	wait then suspend	180

Figure 5-44 Confirming the disk pool changes

5. In the Specify Node panel (Figure 5-45), specify the remote node where the data will be replicated as the mirror copy IASP. In this example, we specify `ITSOSVR2`. Click **Next**.



Configure Geographic Mirroring - Specify Node

Specify the node on which to configure remote disk pool mirroring. You can use the host name or the ip address

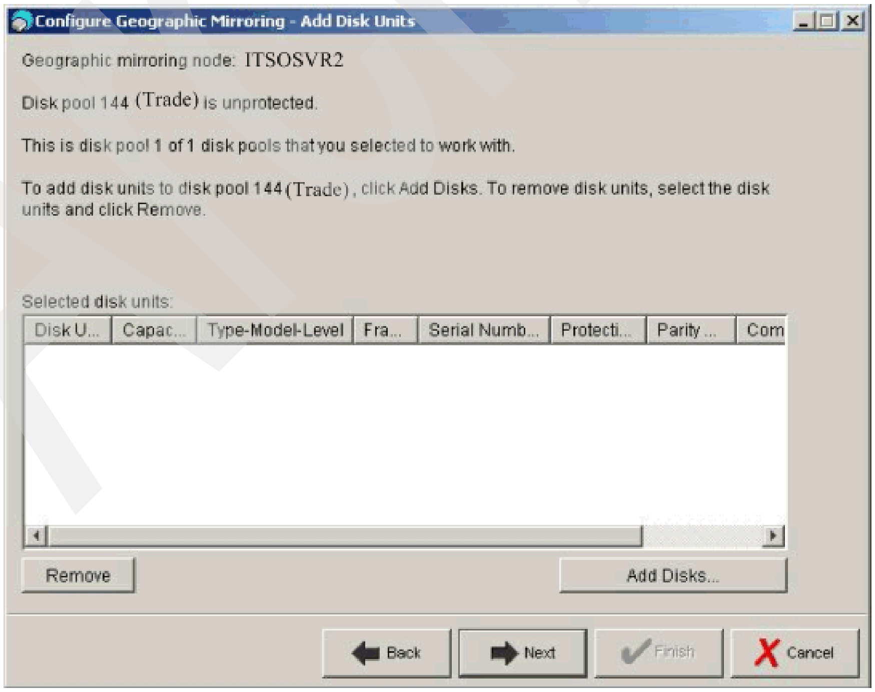
Remote Node: ITSOSVR2

Note: To create disk pools for geographic mirroring on the selected node, you will be prompted to sign on to the node with the same authority you use for creating disk pools on System Rchaspst. You sill need to sign on to the remote system and to the service tools server.

Figure 5-45 Specifying the remote node for mirroring

6. In the Add Disk Units panel (Figure 5-46), configure the remote node disk pool on backup node ITSOSVR2.

   a. Click **Add Disks** to specify the non-configured disk units that we previously made available for XSM support on ITSOSVR2.



Configure Geographic Mirroring - Add Disk Units

Geographic mirroring node: ITSOSVR2

Disk pool 144 (Trade) is unprotected.

This is disk pool 1 of 1 disk pools that you selected to work with.

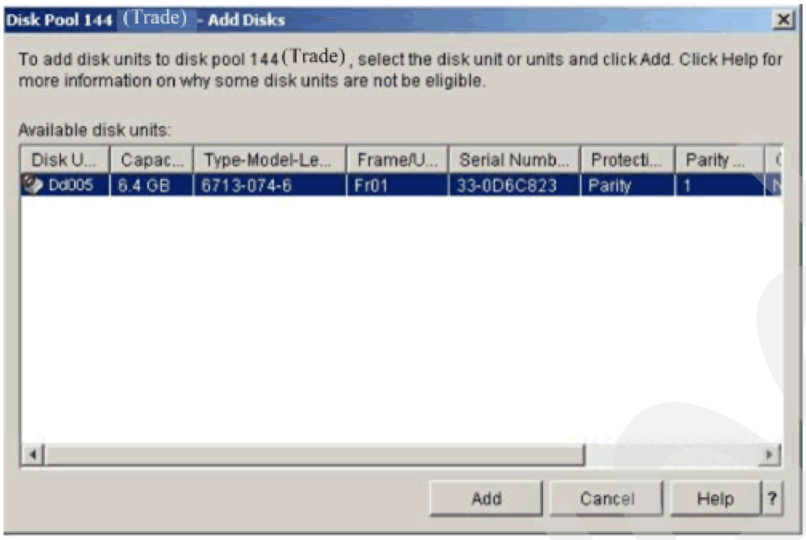
To add disk units to disk pool 144 (Trade) , click Add Disks. To remove disk units, select the disk units and click Remove.

Selected disk units:

Disk U...	Capac...	Type-Model-Level	Fra...	Serial Numb...	Protecti...	Parity ...	Com

Figure 5-46 Configuring the remote node ITSOSVR2 disk pool for Trade

b. In the Add Disks window (Figure 5-47), select the nonconfigured disk and click **Add**.



Disk Pool 144 (Trade) - Add Disks

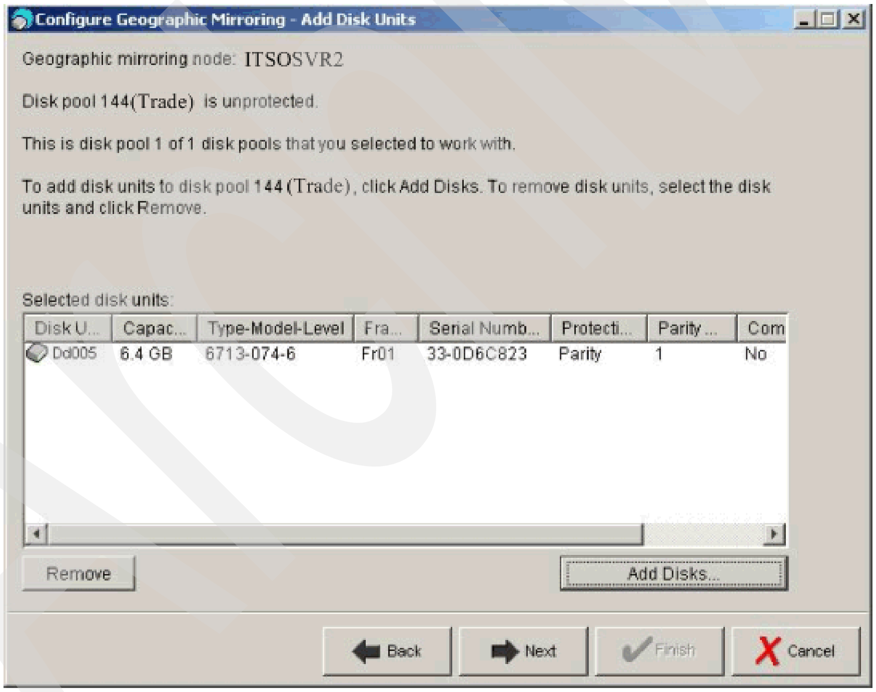
To add disk units to disk pool 144 (Trade), select the disk unit or units and click Add. Click Help for more information on why some disk units are not be eligible.

Available disk units:

Disk U...	Capac...	Type-Model-Le...	Frame/U...	Serial Numb...	Protecti...	Parity ...
Dd005	6.4 GB	6713-074-6	Fr01	33-0D6C823	Parity	1

Figure 5-47 Selecting the nonconfigured disk for the remote disk pool

c. Back in the Add Disk Units panel (Figure 5-48), you should see the nonconfigured remote disk. Confirm its attributes for remote disk pool creation. Click **Next**.



Configure Geographic Mirroring - Add Disk Units

Geographic mirroring node: ITSOSVR2

Disk pool 144(Trade) is unprotected.

This is disk pool 1 of 1 disk pools that you selected to work with.

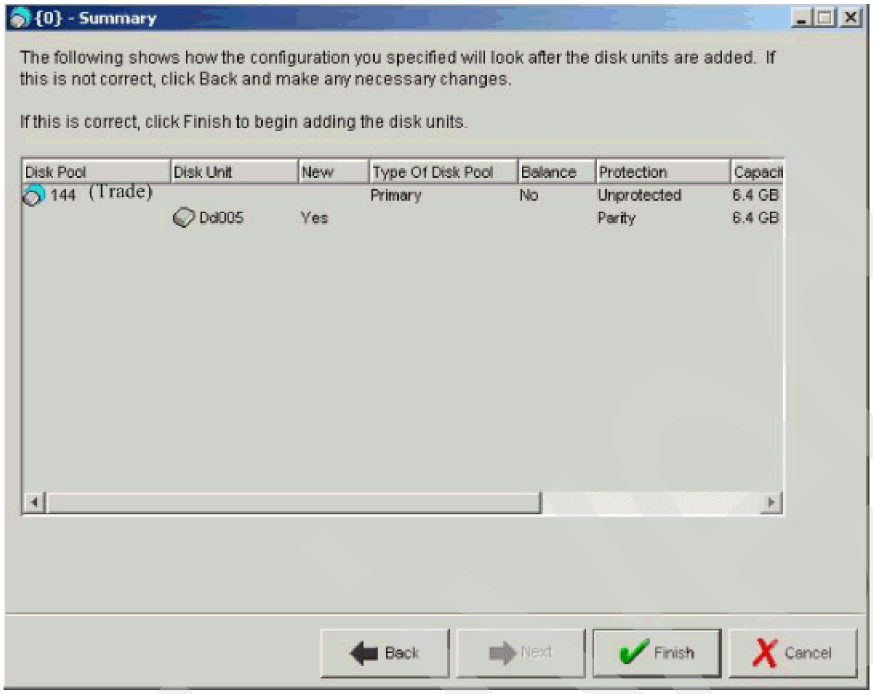
To add disk units to disk pool 144 (Trade), click Add Disks. To remove disk units, select the disk units and click Remove.

Selected disk units:

Disk U...	Capac...	Type-Model-Level	Fra...	Serial Numb...	Protecti...	Parity ...	Com
Dd005	6.4 GB	6713-074-6	Fr01	33-0D6C823	Parity	1	No

Figure 5-48 Verifying the nonconfigured disk for the remote disk pool

7. In the Summary panel (Figure 5-49), review the remote disk pool creation summary information, and click **Finish**.



The following shows how the configuration you specified will look after the disk units are added. If this is not correct, click Back and make any necessary changes.

If this is correct, click Finish to begin adding the disk units.

Disk Pool	Disk Unit	New	Type Of Disk Pool	Balance	Protection	Capaci
144 (Trade)			Primary	No	Unprotected	6.4 GB
	Dd005	Yes			Parity	6.4 GB

Figure 5-49 Summary information

8. A status window opens and indicates that the remote disk pool is being added. This action takes a while, so be patient.

   You see a confirmation message indicating a successful remote pool creation. Click **OK**.

At this point, geographic mirroring is complete as illustrated in Figure 5-50.

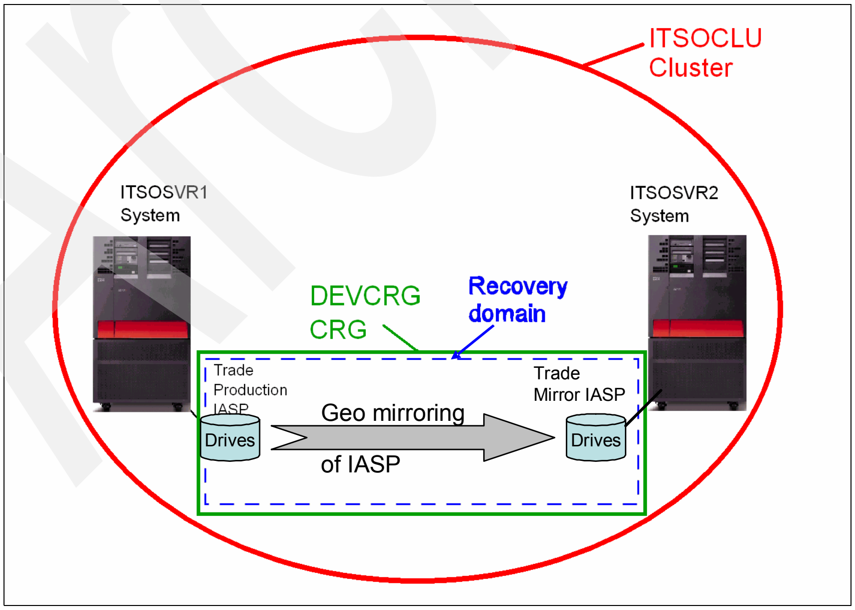


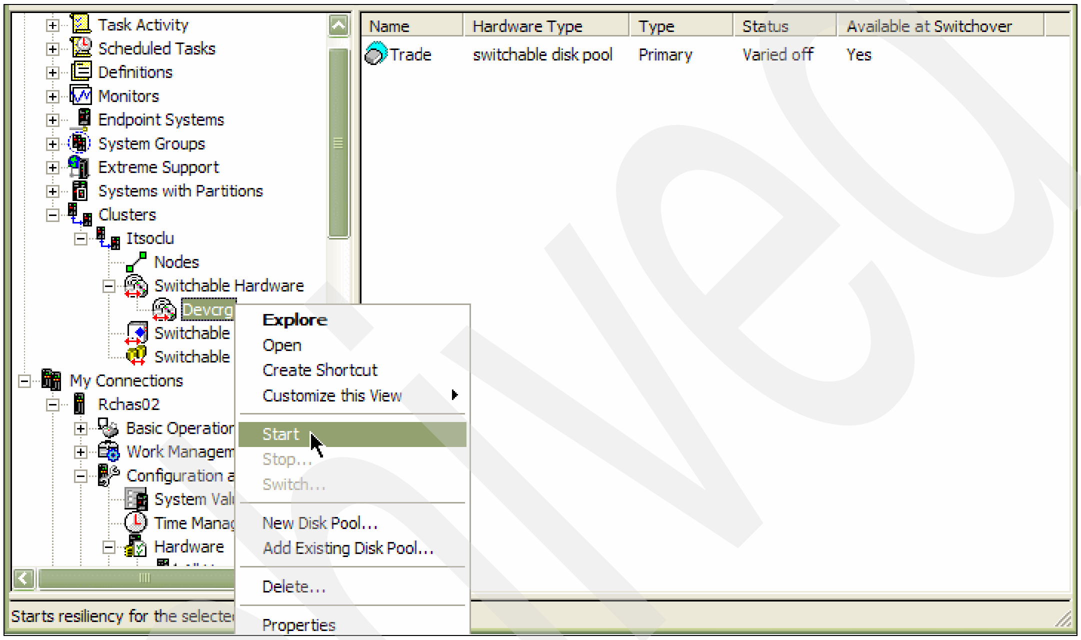
Figure 5-50 Current configuration after geographic mirroring is enabled

In the next section, you learn how to start geographic mirroring and perform a switchover from the primary node to the backup node.

5.3.7 Making the geographic mirror available

Now that you have configured XSM, you need to start it and make available for the application.

1. In iSeries Navigator, as shown in Figure 5-51, expand **Management Central** → **Clusters** → ***your cluster*** → **Switchable Hardware**. Right-click ***your CRG*** and select **Start**.

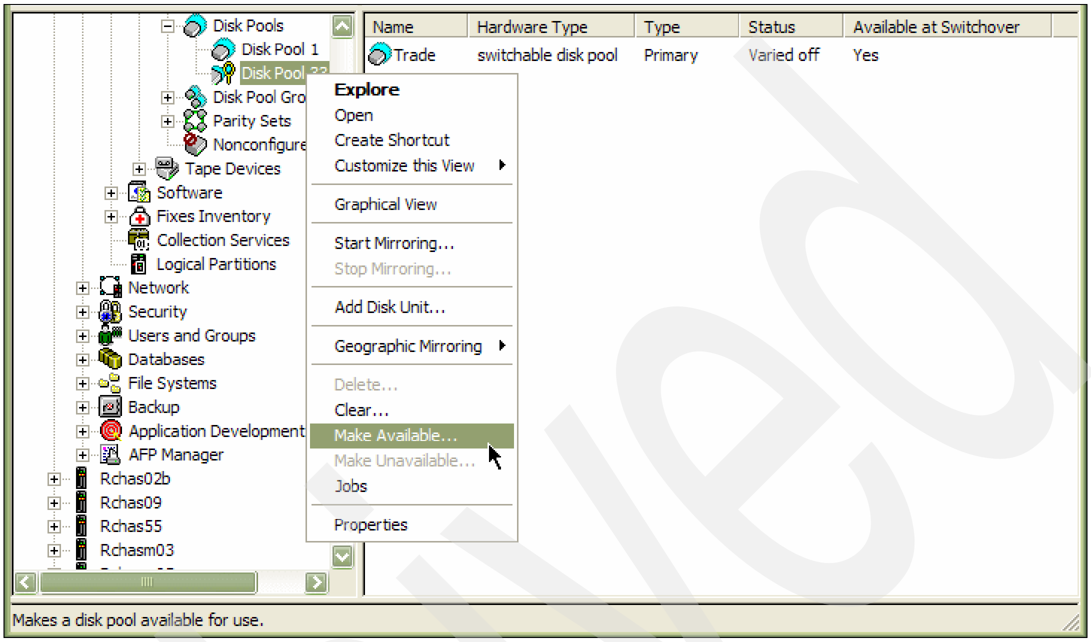


Name	Hardware Type	Type	Status	Available at Switchover
Trade	switchable disk pool	Primary	Varied off	Yes

Starts resiliency for the selecte

Figure 5-51 Starting the CRG

2. Make the disk pool available on the primary node, which in this example is on the node ITSOSVR1. In iSeries Navigator, as shown in Figure 5-52, expand ***your primary system*** → **Configuration and Service** → **Hardware** → **Disk Units** → **Disk Pools**. Then right-click your IASP disk pool and select **Make Available**.



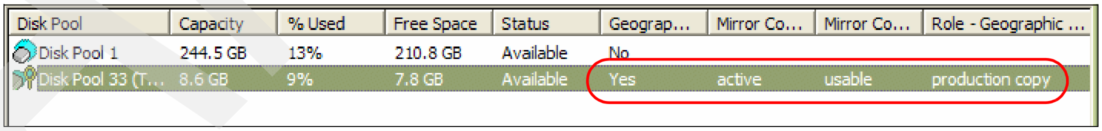
Name	Hardware Type	Type	Status	Available at Switchover
Trade	switchable disk pool	Primary	Varied off	Yes

Makes a disk pool available for use.

Figure 5-52 Varying on the IASP

3. On the window that opens, click **Make Available**.
4. A status window opens. It could take up to several minutes to vary on the disk pool.
5. When the disk pool is varied on, a new window opens, informing you that the action completed successfully. Click **OK**.
6. When XSM is in the operational mode, you should see the status of XSM in iSeries Navigator for your disk pools (see Figure 5-53). However, the first time you start geographic mirroring, any data located on the primary node's IASP is synchronized with the backup node. Depending on the size of the data, this action may take a long time.

   If you don't see the columns shown in Figure 5-53, add them by using **View** → **Customize this view** → **Columns** menu item.



Disk Pool	Capacity	% Used	Free Space	Status	Geograp...	Mirror Co...	Mirror Co...	Role - Geographic ...
Disk Pool 1	244.5 GB	13%	210.8 GB	Available	No			
Disk Pool 33 (T...	8.6 GB	9%	7.8 GB	Available	Yes	active	usable	production copy

Figure 5-53 Disk pools status

7. At this point the data on IASP available for testing. You can run your test application that access the data on the cluster. Use the takeover IP address to access the cluster.

We demonstrate the WebSphere configuration using the switchover IP later in this chapter. See "WebSphere data source changes for the Trade6 application" on page 158 and "WebSphere transaction log changes" on page 162 to learn how to configure WebSphere for geographic mirrored data.

5.3.8 Simulating an XSM switchover

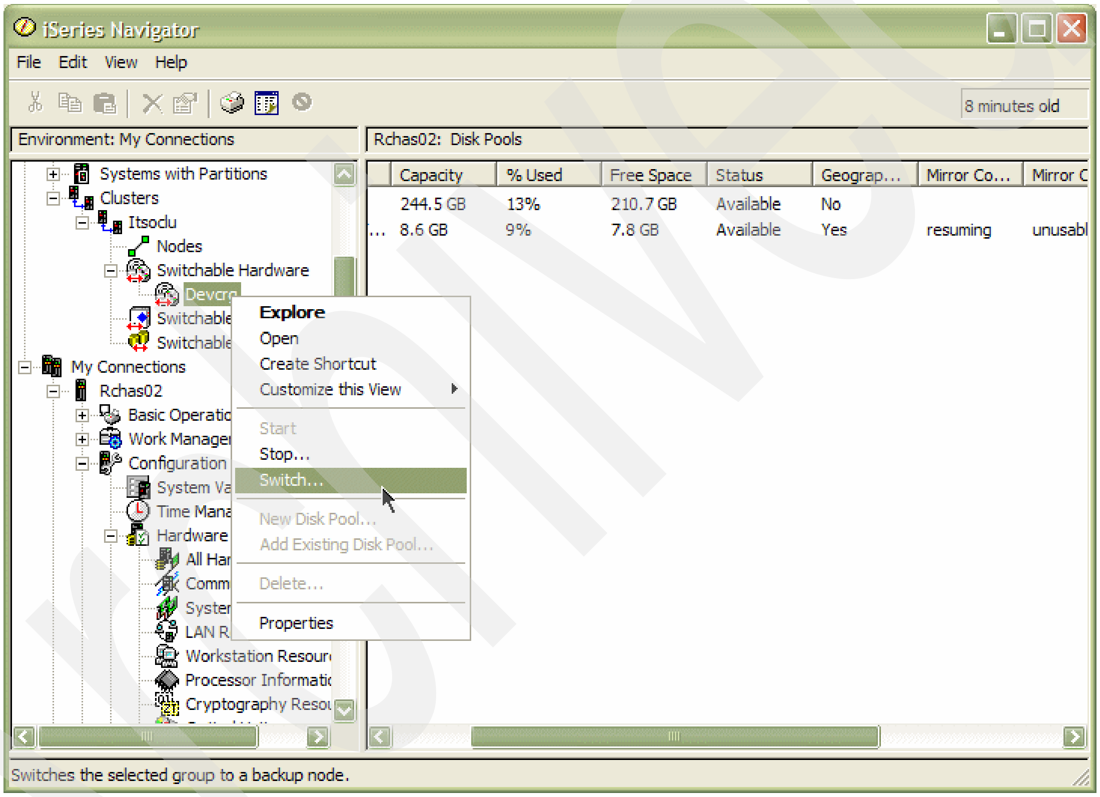
Performing a manual switchover causes the current primary node to switch over to the backup node, as defined in the CRG's recovery domain. When this happens, the current roles of the nodes in the recovery domain of a CRG change so that:

- The current primary node is assigned the role of last active backup.
- The current first backup is assigned the role of primary.
- Subsequent backups are moved up one in the order of backups.

A switchover is only allowed on CRGs that have a status of *Active*.

To perform a switchover on a resource, follow these steps:

1. In iSeries Navigator, as shown in Figure 5-54, expand **Management Central** → **Clusters** → ***your cluster*** → **Switchable Hardware**. Right-click ***your CRG***, and select **Switch**.

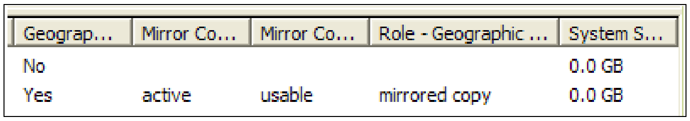


	Capacity	% Used	Free Space	Status	Geograp...	Mirror Co...	Mirror C
	244.5 GB	13%	210.7 GB	Available	No		
...	8.6 GB	9%	7.8 GB	Available	Yes	resuming	unusabl

Switches the selected group to a backup node.

Figure 5-54 Performing a test switch

2. On the confirmation window that opens, click **Yes**.
3. It takes a minimum of several minutes to perform a switch. The switch time depends on a variety of factors.
4. After the switch occurs, look at the status of your disk pools on your primary system (ITSOSVR1 in our example). The status should show that now ITSOSVR1 has a mirror copy of the data (see Figure 5-55). Compare it with the status in Figure 5-53.



Geograp...	Mirror Co...	Mirror Co...	Role - Geographic ...	System S...
No				0.0 GB
Yes	active	usable	mirrored copy	0.0 GB

Figure 5-55 Disk pool status

5. Check the status of the disk pools on the backup node (ITSOSVR2 in our example). The status should show that it now holds the production copy of the data.

At this point, we switch back to the primary node, ITSOSVR1.

This concludes the configuration process of XSM for our application data. At this point, you can start using the database that is protected by XSM.

5.3.9 Special case: Manual failover

If the server that is hosting the primary cluster node fails due to a hard outage caused by a high-impact hardware or operating system failure, that node will likely go into a partition state. To recover from that failure and to enable users to access the independent disk pool again on the remaining cluster node, you should perform a manual switchover. You can find the detailed instructions for doing a manual switchover in “Recovery steps for a two-node geographic mirroring configuration” on page 293.

5.3.10 Green-screen commands

Green-screen commands for monitoring and managing cluster and CRGs are conveniently available. These commands will be used later to simulate a failover situation to get the primary cluster node to switch to the backup cluster node. The commands presented in the following sections can help you with cluster and CRG monitoring and management.

Cluster information

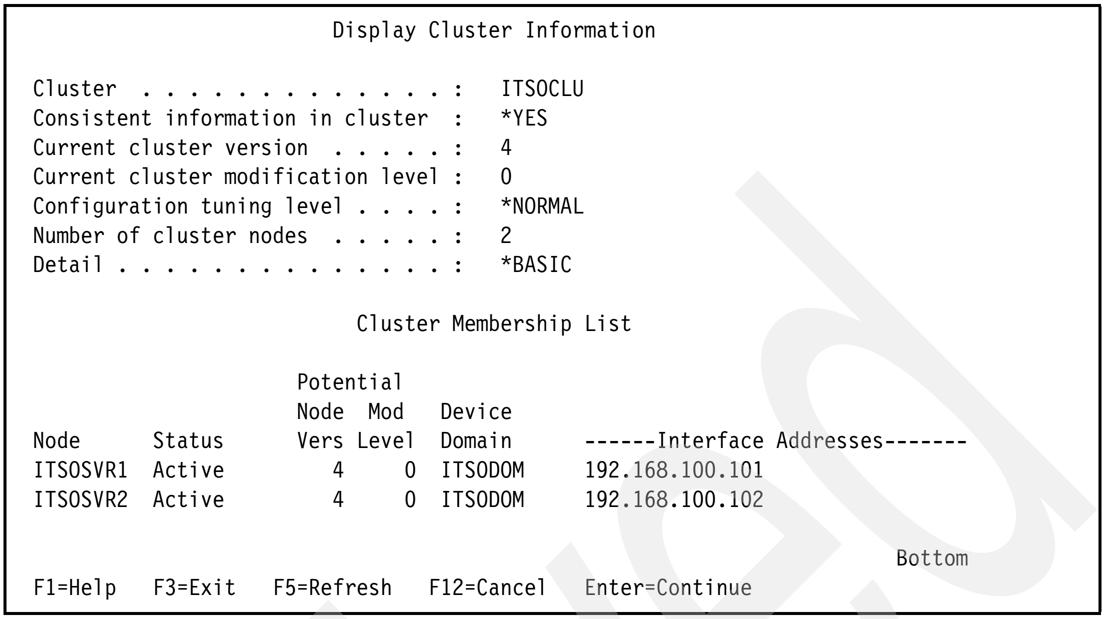
Use the Display Cluster Information (DSPCLUINF) command to display or print information about a cluster. You must invoke it from a node in the cluster.

The information displayed or printed may not be current if the command is called on a node that has a status of Inactive or Failed. In this case, the information that is displayed or printed will reflect the state of the cluster when the node was last active. You can use this command to either display or print basic information describing the cluster membership list or complete configuration information about the cluster. This command may be called from a CRG exit program.

To display the information about the ITSOCLU cluster that you created, use the DSPCLUINF command with the cluster parameter:

```
DSPCLUINF CLUSTER(ITSOCLU)
```

The DSPCLUINF command displays information about, and the status of, the cluster node as shown in Figure 5-56.



```
                         Display Cluster Information

 Cluster  . . . . . . . . . . . . . :   ITSOCLU
 Consistent information in cluster  :   *YES
 Current cluster version  . . . . . :   4
 Current cluster modification level :   0
 Configuration tuning level . . . . :   *NORMAL
 Number of cluster nodes  . . . . . :   2
 Detail . . . . . . . . . . . . . . :   *BASIC

                           Cluster Membership List

                      Potential
                      Node  Mod   Device
 Node      Status     Vers Level  Domain      ------Interface Addresses-------
 ITSOSVR1  Active        4     0  ITSODOM     192.168.100.101
 ITSOSVR2  Active        4     0  ITSODOM     192.168.100.102

                                                                         Bottom
 F1=Help   F3=Exit   F5=Refresh   F12=Cancel   Enter=Continue
```

Figure 5-56   Display Cluster Information display to view cluster status

Cluster resource group information

Use the Display Cluster Resource Group Information (DSPCRGINF) command to display or print information about CRGs. You must invoke it from a node in the cluster. The information that is displayed or printed may not be current if the command is called on a node that has a status of *Inactive* or *Failed*. In this case, the information that is displayed or printed will reflect the state of the cluster when the node was last active. You can use this command to display or print a list of CRGs or to complete information about a CRG.

If the CRG(*LIST) option is specified, the request for information is not distributed to other nodes in the cluster. The information about the CRGs shows the values obtained from the node running this command. Several conditions, such as Cluster Resource Services not being active on the node running the command, may produce inconsistent information about a CRG in the cluster.

When you request information for a specific CRG, basic information for the CRG is always shown. In addition, you can request additional details that include the recovery domain and the list of resilient devices. If Cluster Resource Services has been started, this command returns information about the CRG even if it does not exist on the node from which the command is called provided at least one recovery domain node is active.

If Cluster Resource Services has not been started, then the following actions may occur:

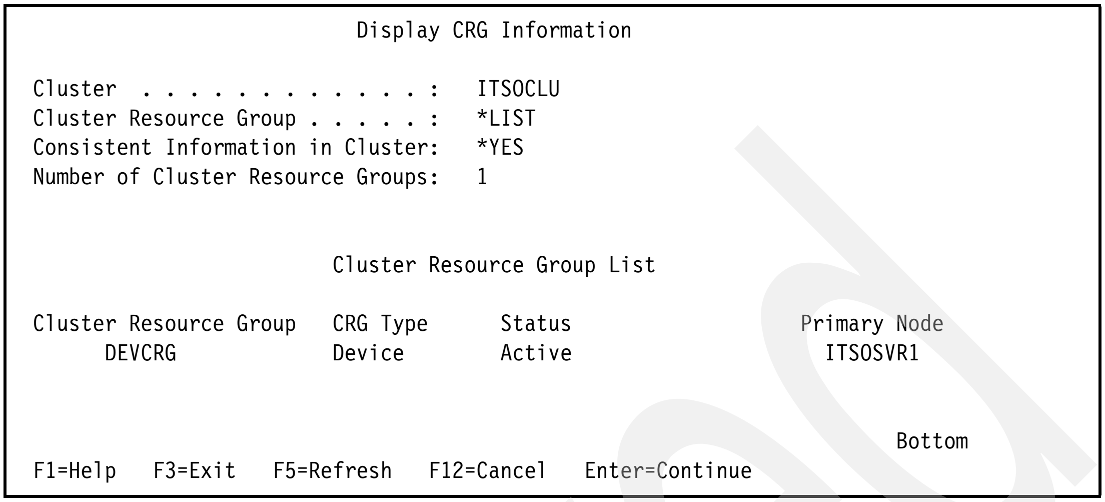
- The information that is returned may not be current.
- Information is returned only for a CRG that exists on the node running the command.

> **Note:** This command may be called from a CRG exit program. However if the CRG exit program was called as a result of the Create Cluster Resource Group (CRTCRG) command and you are requesting information for that CRG, the command will fail.

When you run this command against the ITSOCLU cluster, without specifying the device resource, you see all of the device resource groups defined for the cluster and their respective status (see Figure 5-57).

```
DSPCRGINF CLUSTER(ITSOCLU)
```

Use this command to quickly and easily determine the status of the device resource group.



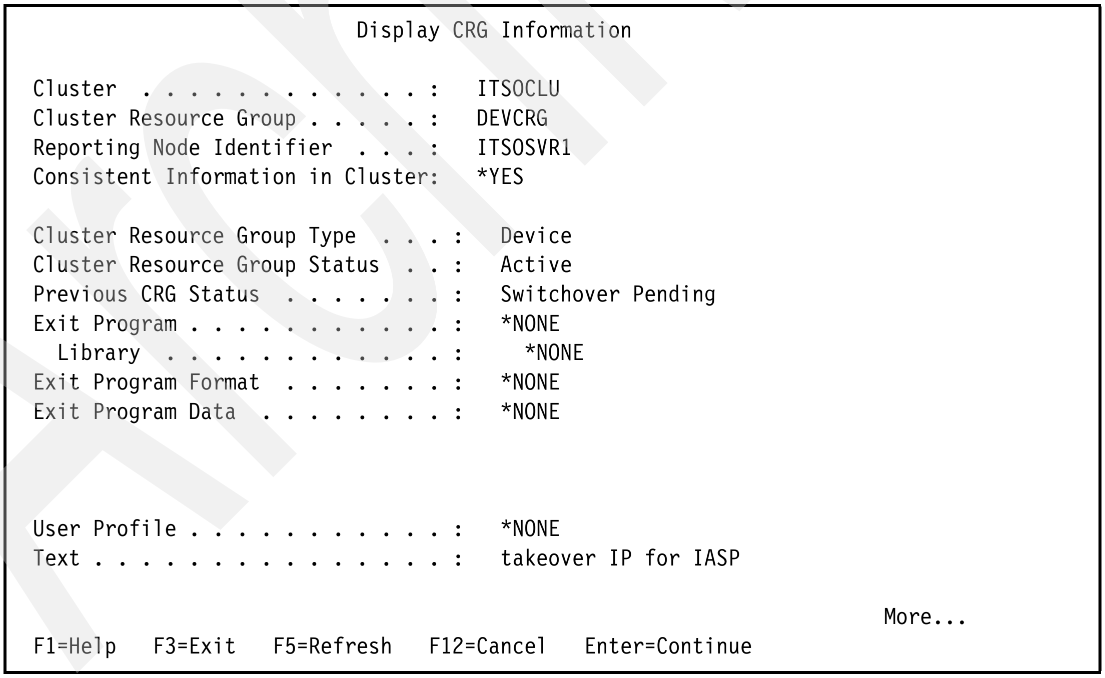
```
                             Display CRG Information

 Cluster  . . . . . . . . . . . . :   ITSOCLU
 Cluster Resource Group . . . . . :   *LIST
 Consistent Information in Cluster:   *YES
 Number of Cluster Resource Groups:   1


                           Cluster Resource Group List

 Cluster Resource Group   CRG Type      Status                   Primary Node
       DEVCRG             Device        Active                     ITSOSVR1



                                                                         Bottom
 F1=Help   F3=Exit   F5=Refresh   F12=Cancel   Enter=Continue
```

Figure 5-57 Display CRG Information panel

To see more detailed information about the device resource group and the status of the group node members, use the CRG specification parameter.

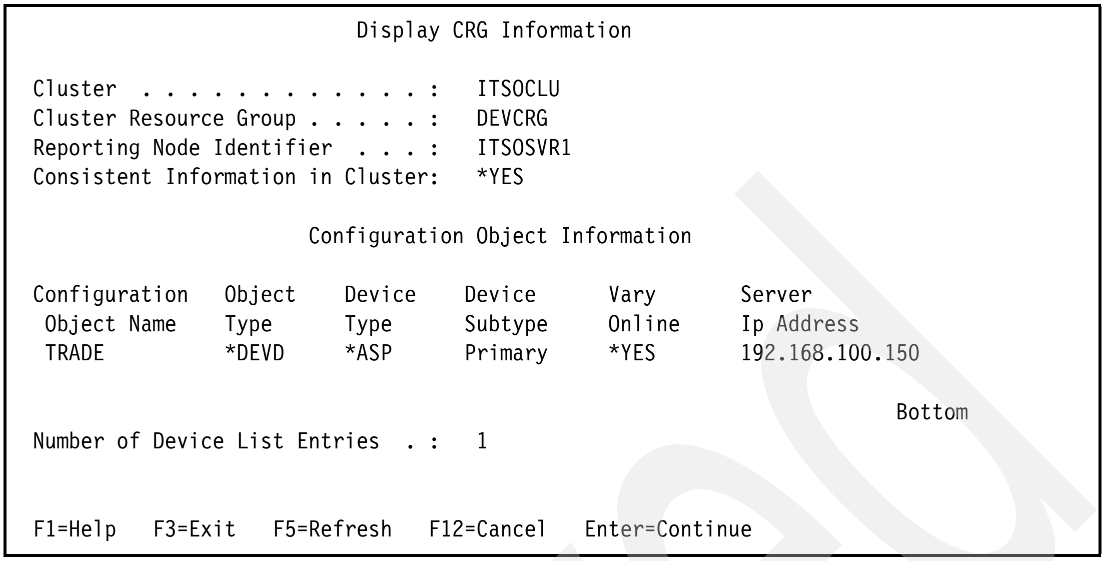
```
DSPCRGINF CLUSTER(ITSOCLU) CRG(DEVCRG)
```

Additional information is displayed with the node status (see Figure 5-58, Figure 5-59, and Figure 5-60). This is a helpful command to use during failover simulation.



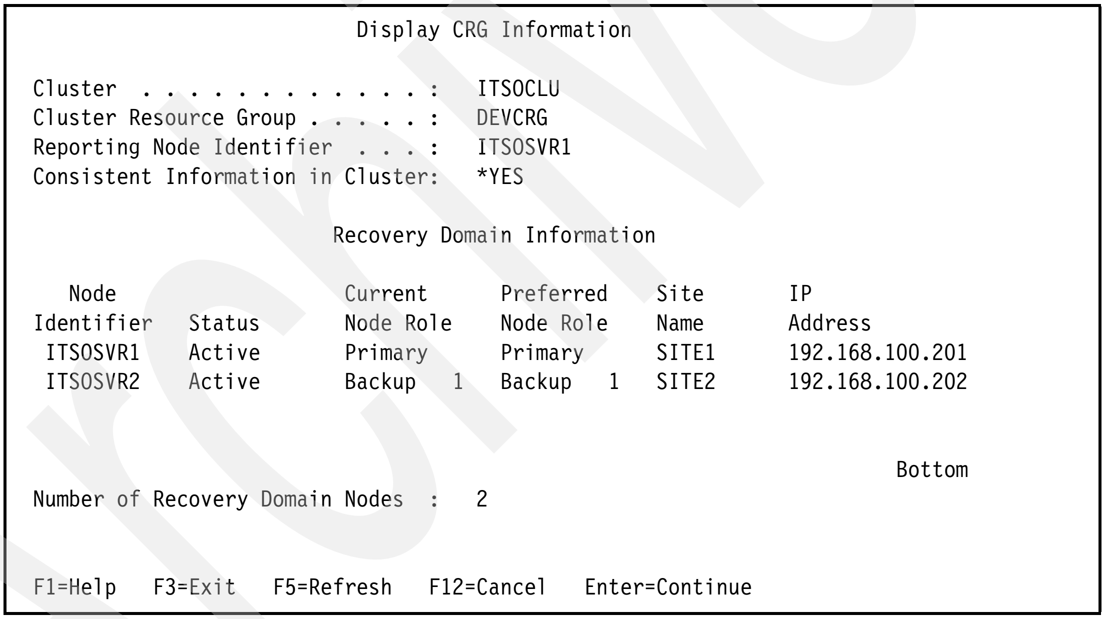
```
                             Display CRG Information

 Cluster  . . . . . . . . . . . . :   ITSOCLU
 Cluster Resource Group . . . . . :   DEVCRG
 Reporting Node Identifier  . . . :   ITSOSVR1
 Consistent Information in Cluster:   *YES

 Cluster Resource Group Type  . . . :   Device
 Cluster Resource Group Status  . . :   Active
 Previous CRG Status  . . . . . . . :   Switchover Pending
 Exit Program . . . . . . . . . . . :   *NONE
   Library  . . . . . . . . . . . . :     *NONE
 Exit Program Format  . . . . . . . :   *NONE
 Exit Program Data  . . . . . . . . :   *NONE




 User Profile . . . . . . . . . . . :   *NONE
 Text . . . . . . . . . . . . . . . :   takeover IP for IASP

                                                                        More...
 F1=Help   F3=Exit   F5=Refresh   F12=Cancel   Enter=Continue
```

Figure 5-58 Display CRG Information panel at the resource group level



```
                          Display CRG Information

Cluster  . . . . . . . . . . . . :   ITSOCLU
Cluster Resource Group . . . . . :   DEVCRG
Reporting Node Identifier  . . . :   ITSOSVR1
Consistent Information in Cluster:   *YES

                      Configuration Object Information

Configuration   Object     Device     Device      Vary       Server
 Object Name    Type       Type       Subtype     Online     Ip Address
 TRADE          *DEVD      *ASP       Primary     *YES       192.168.100.150

                                                                         Bottom
Number of Device List Entries  . :   1


F1=Help   F3=Exit   F5=Refresh   F12=Cancel   Enter=Continue
```

Figure 5-59   Display CRG Information panel at the resource group level



```
                          Display CRG Information

Cluster  . . . . . . . . . . . . :   ITSOCLU
Cluster Resource Group . . . . . :   DEVCRG
Reporting Node Identifier  . . . :   ITSOSVR1
Consistent Information in Cluster:   *YES

                         Recovery Domain Information

   Node                      Current       Preferred     Site       IP
Identifier   Status          Node Role     Node Role     Name       Address
 ITSOSVR1    Active          Primary       Primary       SITE1      192.168.100.201
 ITSOSVR2    Active          Backup    1   Backup    1   SITE2      192.168.100.202


                                                                         Bottom
Number of Recovery Domain Nodes  :   2


F1=Help   F3=Exit   F5=Refresh   F12=Cancel   Enter=Continue
```

Figure 5-60   Display CRG Information panel at the resource group level

For other CL commands for geographic mirroring, see the iSeries Information Center at:

http://publib.boulder.ibm.com/infocenter/iseries/v5r3/index.jsp?lang=en

5.4 Firewall configuration

Configuring the firewall is similar to the configuration that is described in 4.3.2, “Firewall configuration” on page 65. The only difference is the size of an appliance. You should use Cisco PIX 535 as a high-performance, purpose-built security appliance instead of Cisco PIX 506E. All of these appliances run the same software, so there is no difference in how you set up the rules.

5.5 Configuring the HTTP and WebSphere servers

Now that we configured the iSeries cluster and XSM, we can configure WebSphere and the HTTP servers, as well as install the sample application. At the end of this process, we should run a test to verify the proper configuration.

5.5.1 Installing the Web server plug-in

In this scenario we take advantage of the integrated HTTP Server (powered by Apache). To establish a connection between the Web servers and application servers, you need to install additional software, called the *Web server plug-in*. The Web server plug-in is a feature of WebSphere Application Server V6. The Web server plug-in installation is described in Appendix B, "Installing WebSphere Application Server Version 6" on page 255.

After you install the Web server plug-in on both WEBSERVER1 and WEBSERVER2, the following Integrated File System (IFS) directories exist:

- /QIBM/ProdData/WebSphere/AppServer/V6/ND
- /QIBM/UserData/WebSphere/AppServer/V6/ND
- /QIBM/UserData/WebSphere/AppServer/V6/service

On each iSeries, the QWAS6 library is created. This library is the default location for the Web server plug-in program.

5.5.2 Creating HTTP Server instances

This scenario for a large customer uses two HTTP server instances. Both of these instances work in parallel. Load Balancer is responsible for routing users request to the appropriate Web server instance. Using two Web server instances gives both scalability and high availability to your Web site.

Table 5-5 contains the parameters that are required for creating HTTP Server instances.

Table 5-5 HTTP Server instances parameters

Instance name	Host name	TCP/IP address	TCP/IP Port	Instance root directory
WebSvr1	WEBSERVER1	1.1.1.200	80	/www/WebSvr1
WebSvr2	WEBSERVER2	1.1.1.201	80	/www/WebSvr2

To create the HTTP server instances, use the IBM Web Administration for iSeries GUI. You can learn how to create a Web server instance in "Creating an HTTP Server instance" on page 72. Using the IBM Web Administration GUI, create an HTTP server instance on each Web server iSeries, using the parameters shown in Table 5-5.

5.5.3 Configuring a cluster of WebSphere servers

Configuring a cluster of WebSphere servers is not a trivial task. You must follow the sequence of the steps as presented in this section. See Figure 5-3 on page 98 for a overview of the configuration.

Creating HTTP Server profiles

After creating a Web server instances on both the WEBSERVER1 and WEBSERVER2 iSeries, you must create HTTP server WebSphere profiles on each of these iSeries. The main reason to create an HTTP server profile is that during the creation, an IFS directory structure, similar to a WebSphere profile, is also created. The HTTP Server instance looks for the plugin-cfg.xml file in this newly created directory structure. Log files related to the plug-in component are also generated within this directory structure.

One of two parameters that is needed to create an HTTP Server profile is the profile name. A good practice for naming this parameter is to use the same name that was assigned to the application server profile. For this scenario, specify the name that corresponds to the *deployment manager profile*. Although the deployment manager profile has not yet been created, the profile name that is used in this scenario is listed in Table 5-6. The name of the deployment manager profile is *DmgrSvr*.

Complete the following steps to create an HTTP server profile:

1. Log on to the *WEBSERVER1* iSeries, and enter the Start Qshell (`STRQSH`) command.
2. On the Qshell command line, type the following command:

   ```
   cd /QIBM/ProdData/WebSphere/AppServer/V6/ND/bin
   ```

3. Run the wasprofile script. For this scenario, the command is:

   ```
   wasprofile -create -profileName DmgrSvr -templatePath http
   ```

   > **Note:** The templatePath parameter assumes the specified template is in the *install_root*/profileTemplates directory. In this scenario, the *http* profile template is found in the /QIBM/ProdData/WebSphere/AppServer/V6/ND/profileTemplates directory.

   After the script is invoked, you see a message similar to the one shown in Example 5-1 in the Qshell session.

   Example 5-1 Web server profile creation output

   ```
   > wasprofile -profileName DmgrSvr -templatePath http
   INSTCONFSUCCESS: Success: The profile now exists.
     $
   ```

4. To create the HTTP server profile on the WEBSERVER2 iSeries, log on to the WEBSERVER2 iSeries and repeat steps 2 and 3.

After creating the HTTP server profiles, a directory structure is created on each iSeries Web server system. The directory structure on each iSeries is /QIBM/UserData/WebSphere/AppServer/V6/ND/profiles/*profile-name*, where *profile-name* for both the WEBSERVER1 and WEBSERVER2 iSeries is DmgrSvr.

In future steps, the plug-in configuration file is moved to these newly created directories on the WEBSERVER1 and WEBSERVER2 iSeries. For now, continue with creating the deployment manager profile.

Creating deployment manager profile

Although you can create the deployment manager on either iSeries, which resides in the private LAN, for this scenario, we create the profile on the APPSERVER1 iSeries. Table 5-6 contains the parameters that are used to create the WebSphere Application Server deployment manager profile.

Table 5-6 WebSphere Application Server deployment manager profile parameters

Profile name	Host name	Starting port
DmgrSvr	APPSERVER1	30000

The wasprofile script is executed within Qshell to create this profile. To run this script, the iSeries user profile must have *ALLOBJ authority. Complete the following steps:

1. Log on to the APPSERVER1 iSeries.
2. On the OS/400 command line, enter the `STRQSH` command.
3. On the Qshell command line, type the following command:

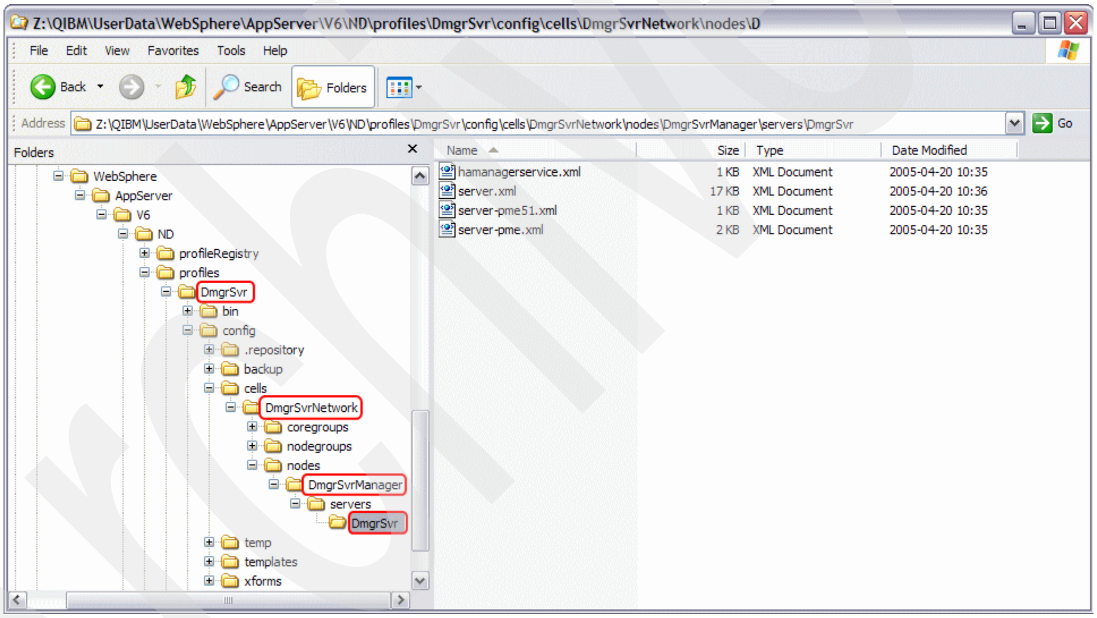
   ```
   cd /QIBM/ProdData/WebSphere/AppServer/V6/ND/bin
   ```

4. Run the wasprofile script to create the deployment manager profile. For this scenario, the command is:

   ```
   wasprofile -create -profileName DmgrSvr -templatePath dmgr -startingPort 30000
   ```

   > **Note:** The templatePath parameter assumes the specified template is in the *install_root*/profileTemplates directory. In this scenario, the dmgr profile template is in the /QIBM/ProdData/WebSphere/AppServer/V6/ND/profileTemplates directory.

   The result of running the wasprofile script is the creation of a new directory structure as shown in Figure 5-61.


Z:\QIBM\UserData\WebSphere\AppServer\V6\ND\profiles\DmgrSvr\config\cells\DmgrSvrNetwork\nodes\D

Address: Z:\QIBM\UserData\WebSphere\AppServer\V6\ND\profiles\DmgrSvr\config\cells\DmgrSvrNetwork\nodes\DmgrSvrManager\servers\DmgrSvr

Name	Size	Type	Date Modified
hamanagerservice.xml	1 KB	XML Document	2005-04-20 10:35
server.xml	17 KB	XML Document	2005-04-20 10:36
server-pme51.xml	1 KB	XML Document	2005-04-20 10:35
server-pme.xml	2 KB	XML Document	2005-04-20 10:35

Figure 5-61 Deployment manager profile directory structure

   As you can see, in Figure 5-61, the root directory name for the profile is the same as the deployment manager profile's name (DmgrSvr). The cell's name consists of the deployment manager profile name, followed by the *Network* suffix (DmgrSvrNetwork). The node's name consists of the deployment manager profile name followed by the *Manager* suffix (DmgrSvrManager).

   Figure 5-61 also shows one application server created under the cell's directory structure. The name of the server is the same as the deployment manager profile name (DmgrSvr). This server is responsible for administration of the entire cell.

5. To start the new WebSphere Application Server profile, return to the Qshell session on the APPSERVER1 iSeries. Type the following command:

   ```
   startServer -profileName DmgrSvr
   ```

After the server is started, you see messages similar to those in Example 5-2 in the Qshell session.

Example 5-2   Starting deployment manager server

```
> startServer -profileName DmgrSvr
   CPC1221: Job 035382/QEJBSVR/DMGRSVR submitted to job queue QWASJOBQ in
   library QWAS6.
   WAS6123: Application server started.
    Cause . . . . . :   Application server DmgrSvr in profile (instance)
      DmgrSvr has started and is ready to accept connections on admin port
      30000.
   $
```

As a result of using the startServer command, one job in the QWAS6 subsystem is started. The name of the job is the same as the name of the deployment manager profile (DmgrSvr).

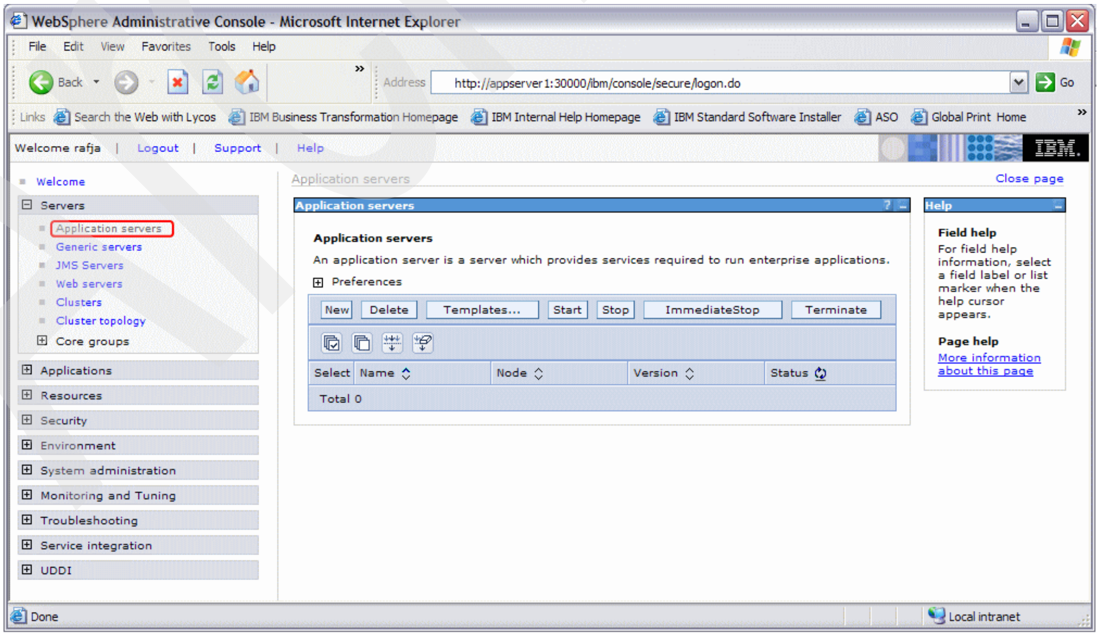
> **Important:** The admin port for the WebSphere Application Server profile is shown in the output for the startServer command. This port is required when opening the WebSphere Administrative Console. As shown in Example 5-2, the admin port is 30000.

6. Start the deployment manager WebSphere Administrative Console by entering the following Uniform Resource Locator (URL) in a browser:

   `http://APPSERVER1:30000/ibm/console`

   By default, WebSphere Administration Console security is turned off which means that only a user ID needs to be specified. Specify any user ID.

   Figure 5-64 shows the cell’s WebSphere Administrative Console. At this time, no application servers have yet been created. Application servers, which are the engines that serve clients requests, are created in a future step.


WebSphere Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://appserver1:30000/ibm/console/secure/logon.do
Go
Links Search the Web with Lycos IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer ASO Global Print Home
Welcome rafja | Logout | Support | Help
IBM.
Welcome
Servers
Application servers
Generic servers
JMS Servers
Web servers
Clusters
Cluster topology
Core groups
Applications
Resources
Security
Environment
System administration
Monitoring and Tuning
Troubleshooting
Service integration
UDDI
Application servers
Close page
Help
New
Delete
Templates...
Start
Stop
ImmediateStop
Terminate
Done
Local intranet

Application servers

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

⊞ Preferences

Select	Name	Node	Version	Status
Total 0				

Field help

For field help information, select a field label or list marker when the help cursor appears.

Page help

More information about this page

Figure 5-62   Deployment manager console

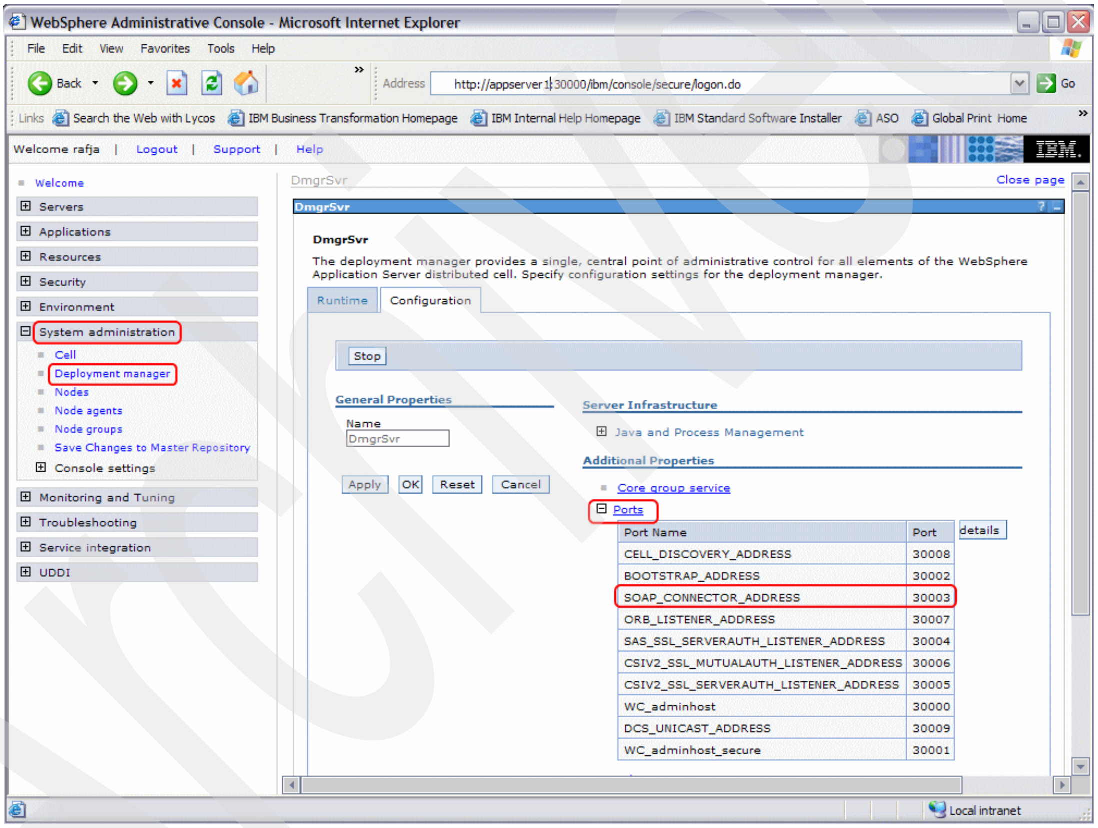
7. Check the deployment manager Simple Object Access Protocol (SOAP) connector address.

   Before proceeding with configuration, check the SOAP connector port. This port acts as a socket to the application server. It is needed later, when the application server is federated to the cell.

   a. To check the SOAP connector port, expand **System administration** and click **Deployment manager**.

   b. On the right, you see the DmgrSvr configuration pane (Figure 5-63). Under Additional Properties, expand **Ports** to see the details of the port.

> **Important:** Before proceeding further, write down the SOAP connector address, as shown in Figure 5-63.


WebSphere Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://appserver1:30000/ibm/console/secure/logon.do
Go
Links Search the Web with Lycos IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer ASO Global Print Home
Welcome rafja | Logout | Support | Help
IBM.
Welcome
Servers
Applications
Resources
Security
Environment
System administration
Cell
Deployment manager
Nodes
Node agents
Node groups
Save Changes to Master Repository
Console settings
Monitoring and Tuning
Troubleshooting
Service integration
UDDI
DmgrSvr
Close page
DmgrSvr
Runtime
Configuration
Stop
General Properties
Server Infrastructure
Java and Process Management
Additional Properties
Core group service
Ports
details
Apply
OK
Reset
Cancel
Local intranet

DmgrSvr

The deployment manager provides a single, central point of administrative control for all elements of the WebSphere Application Server distributed cell. Specify configuration settings for the deployment manager.

Name
DmgrSvr

Port Name	Port
CELL_DISCOVERY_ADDRESS	30008
BOOTSTRAP_ADDRESS	30002
SOAP_CONNECTOR_ADDRESS	30003
ORB_LISTENER_ADDRESS	30007
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS	30004
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS	30006
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS	30005
WC_adminhost	30000
DCS_UNICAST_ADDRESS	30009
WC_adminhost_secure	30001

Figure 5-63 Deployment manager SOAP connector address

Creating the application server profiles

Application servers are engines that run enterprise applications and process clients requests to those applications. In this example, which is dedicated to a large company, high availability and load balancing are needed. To achieve this, one application server is not enough. Figure 5-3 on page 98 shows an environment which has two iSeries servers, each configured with an application server profile.

Table 5-7 contains the parameters that are used to create a WebSphere Application Server profile, one on each iSeries.

Table 5-7 WebSphere Application Server profile parameters

Profile name	**Host name**
AppSvr1	APPSERVER1
AppSvr2	APPSERVER2

To create these profiles, follow these steps:

1. Log on to the APPSERVER1 iSeries.
2. On the OS/400 command line, enter the `STRQSH` command.
3. On the Qshell command line, type the following command:

```
cd /QIBM/ProdData/WebSphere/AppServer/V6/ND/bin
```

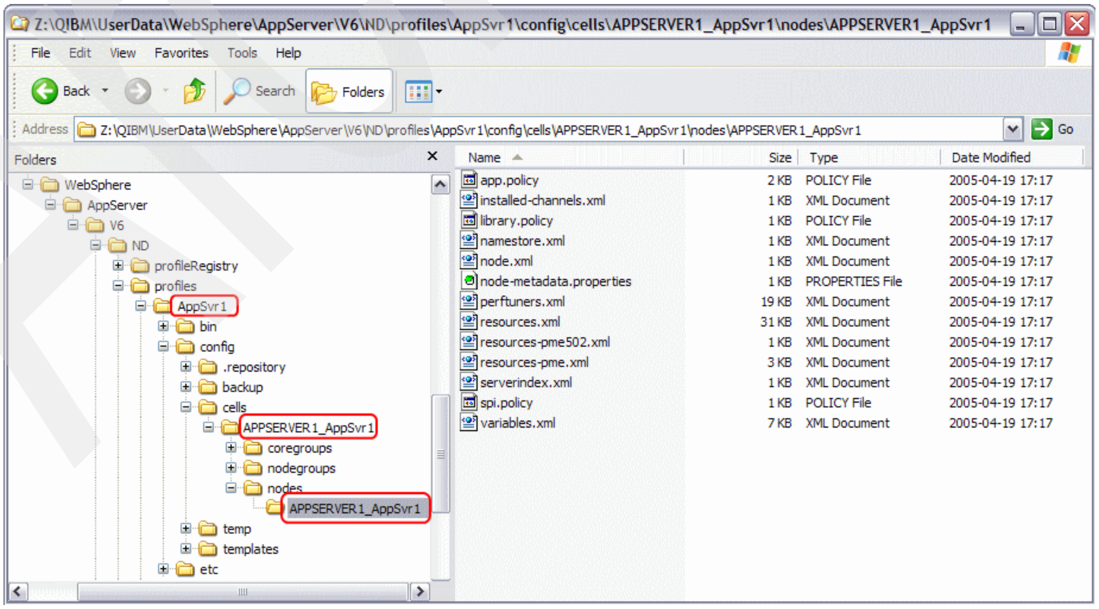
4. Run the wasprofile script to create the application server's managed node. Type the following command:

```
wasprofile -create -profileName AppSvr1 -templatePath managed
```

> **Note:** The templatePath parameter assumes that the specified template is in the *install_root*/profileTemplates directory. In this scenario, the *managed* profile template is found in the /QIBM/ProdData/WebSphere/AppServer/V6/ND/profileTemplates directory.

This command creates a new IBM WebSphere Application profile. Having the template path set to *managed* means that the new profile will be created without an application server. The application server, which handles users requests, will be created later, at the cluster level.

Figure 5-64 shows the directory structure created for the new application profile. The root directory name for the profile is the same as the profile's name that was specified in the **`wasprofile`** command (AppSvr1). This figure also shows that the name of the cell and the name of the node is the same. The name consists of the iSeries system name followed by the profile name. In this case, the name is APPSERVER1_AppSvr1. Notice also that because no application servers were created, no *server* directory is present.


Z:\QIBM\UserData\WebSphere\AppServer\V6\ND\profiles\AppSvr1\config\cells\APPSERVER1_AppSvr1\nodes\APPSERVER1_AppSvr1
File
Edit
View
Favorites
Tools
Help
Back
Search
Folders
Address
Z:\QIBM\UserData\WebSphere\AppServer\V6\ND\profiles\AppSvr1\config\cells\APPSERVER1_AppSvr1\nodes\APPSERVER1_AppSvr1
Go
Folders
WebSphere
AppServer
V6
ND
profileRegistry
profiles
AppSvr1
bin
config
.repository
backup
cells
APPSERVER1_AppSvr1
coregroups
nodegroups
nodes
APPSERVER1_AppSvr1
temp
templates
etc

Name	Size	Type	Date Modified
app.policy	2 KB	POLICY File	2005-04-19 17:17
installed-channels.xml	1 KB	XML Document	2005-04-19 17:17
library.policy	1 KB	POLICY File	2005-04-19 17:17
namestore.xml	1 KB	XML Document	2005-04-19 17:17
node.xml	1 KB	XML Document	2005-04-19 17:17
node-metadata.properties	1 KB	PROPERTIES File	2005-04-19 17:17
perftuners.xml	19 KB	XML Document	2005-04-19 17:17
resources.xml	31 KB	XML Document	2005-04-19 17:17
resources-pme502.xml	1 KB	XML Document	2005-04-19 17:17
resources-pme.xml	3 KB	XML Document	2005-04-19 17:17
serverindex.xml	1 KB	XML Document	2005-04-19 17:17
spi.policy	1 KB	POLICY File	2005-04-19 17:17
variables.xml	7 KB	XML Document	2005-04-19 17:17

Figure 5-64 Profile's directory structure

5. To create the second application server profile, log on to the APPSERVER2 iSeries.
6. On the OS/400 command line, type the STRQSH command.
7. On the Qshell command line, type the following command:

   ```
   cd /QIBM/ProdData/WebSphere/AppServer/V6/ND/bin
   ```

8. Run the wasprofile script to create the application server's managed node. Enter the following command:

   ```
   wasprofile -create -profileName AppSvr2 -templatePath managed
   ```

   After the command completes, a new application server profile is created on the APPSERVER2 iSeries. The directory structure of this new profile is similar to the directory structure shown in Figure 5-64.

At this point two application server profiles are created, one on each iSeries. These profiles are completely separate from one another. To bind them to a common structure, we must *federate* them to a cell.

Federating the application server profiles to the cell

Federating all the distributed application servers in your network, to a cell, allows you to manage these servers better. After federating to a cell, the servers are administered from one central entity called the *deployment manager*. Before adding servers (nodes) to a cell, the deployment manager's host name and SOAP port must be known. For this scenario, refer to "Creating deployment manager profile" on page 141 for the parameters.

To federate each application server profile to the cell, complete the following steps:

1. Log on to the APPSERVER1 iSeries.
2. On the OS/400 command line, type the STRQSH command.
3. On the Qshell command line, type the following command:

   ```
   cd /QIBM/ProdData/WebSphere/AppServer/V6/ND/bin
   ```

4. Run the addNode script to add the application server's managed node to the cell. Type the following command:

   ```
   addNode APPSERVER1 30003 -profileName AppSvr1 -startingport 20050
   ```

   > **Important:** The addNode command creates a node agent. Each node agent uses eleven TCP/IP ports. By specifying the *startingport* parameter, you can select the starting port for these 11 ports.

   After the command completes, the AppSvr1 node is said to be "federated to the cell". Example 5-3 shows the messages that are logged after running the addNode command.

   Example 5-3 AddNode command output messages

   ```
   ADMU0012I: Creating Node Agent configuration for node: APPSERVER1_AppSvr1
   ADMU0014I: Adding node APPSERVER1_AppSvr1 configuration to cell: DmgrSvrNetwork
   ADMU0016I: Synchronizing configuration between node and cell.
   ADMU0018I: Launching Node Agent process for node: APPSERVER1_AppSvr1
   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:
              023735/QEJBSVR/NODEAGENT
   ADMU9990I:
   ADMU0300I: Congratulations! Your node APPSERVER1_AppSvr1 has been successfully
              incorporated into the DmgrSvrNetwork cell.
   ```

To federate the AppSvr2 application server profile, follow these steps:

1. Log on to the APPSERVER2 iSeries.
2. On the OS/400 command line, type the `STRQSH` command.
3. On the Qshell command line, type the following command:

   ```
   cd /QIBM/ProdData/WebSphere/AppServer/V6/ND/bin
   ```

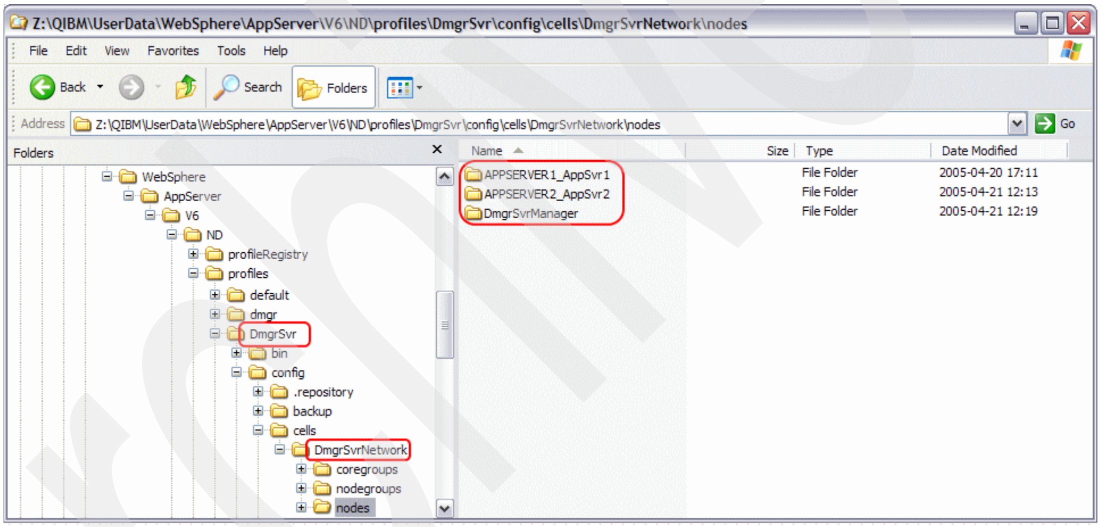
4. Run the addnode script to add the application server's managed node to the cell. Type the following command:

   ```
   addNode APPSERVER2 30003 -profileName AppSvr2 -startingport 20050
   ```

   After the command completes, the AppSvr2 node is said to be "federated to the cell". Example 5-3 shows messages that are logged after running the addNode command.

5. Review the addNode.log, located in the /QIBM/UserData/WebSphere/AppServer/V6/ND/profiles/*profile-name*/logs/addNode.log directory, where *profile-name* is the name of the node added to the cell.

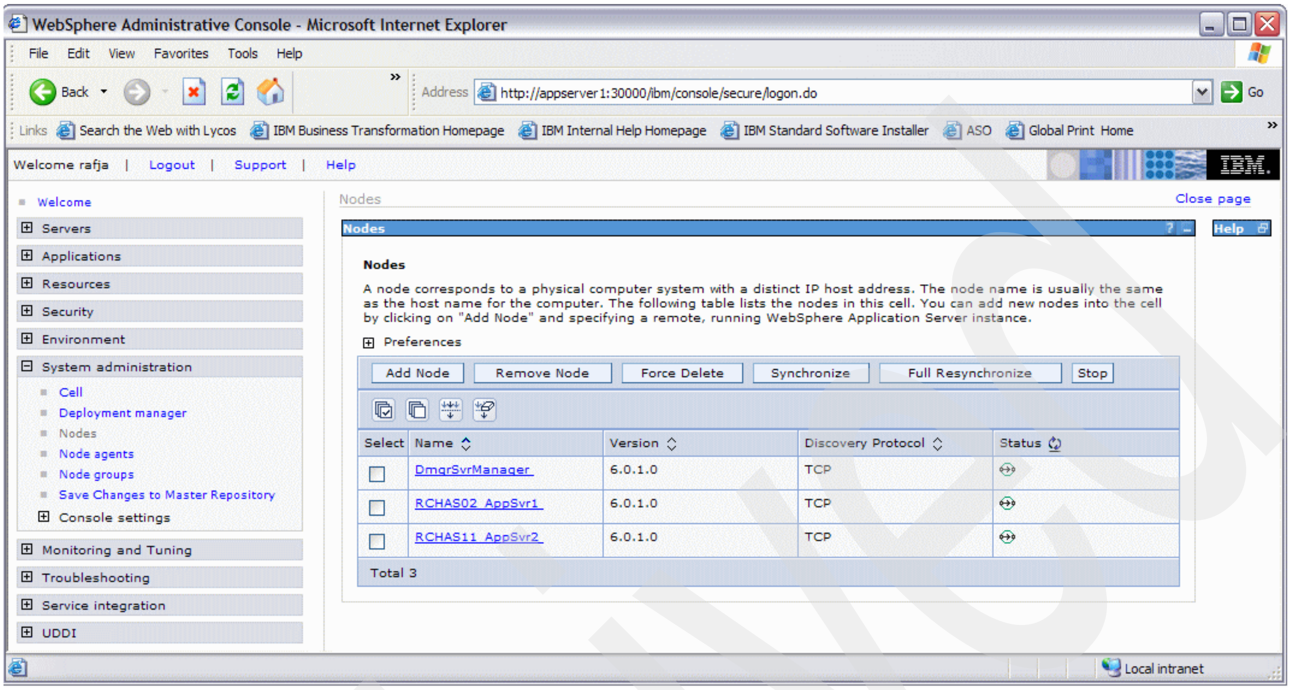
Figure 5-65 shows the directory structure, at the cell level, of all the nodes that are now included in the cell.



Name	Size	Type	Date Modified
APPSERVER1_AppSvr1		File Folder	2005-04-20 17:11
APPSERVER2_AppSvr2		File Folder	2005-04-21 12:13
DmgrSvrManager		File Folder	2005-04-21 12:19

Figure 5-65 Deployment manager directory structure

After all application server nodes are federated into the cell, they are shown in the WebSphere Administrative console of the deployment manager (see Figure 5-66).



Nodes

A node corresponds to a physical computer system with a distinct IP host address. The node name is usually the same as the host name for the computer. The following table lists the nodes in this cell. You can add new nodes into the cell by clicking on "Add Node" and specifying a remote, running WebSphere Application Server instance.

Select	Name	Version	Discovery Protocol	Status
□	DmgrSvrManager	6.0.1.0	TCP	↔
□	RCHAS02_AppSvr1	6.0.1.0	TCP	↔
□	RCHAS11_AppSvr2	6.0.1.0	TCP	↔
Total 3				

Figure 5-66 Application server nodes from the cell's perspective

Now that all the nodes have been federated to the cell, you can create a cluster.

Creating a WebSphere Application Server cluster

Application server clustering allows you to manage the workload of your Java 2 Platform, Enterprise Edition (J2EE), application. A *cluster* is a set of application servers that run the same set of applications. A cluster allows you to achieve better performance because clustering allows more than one application server to process incoming requests.

In this example, the WebSphere Application Server cluster contains four application servers, two servers on each of the two iSeries. This topology demonstrates both vertical and horizontal scaling. You may expand this example to any number of nodes and servers.

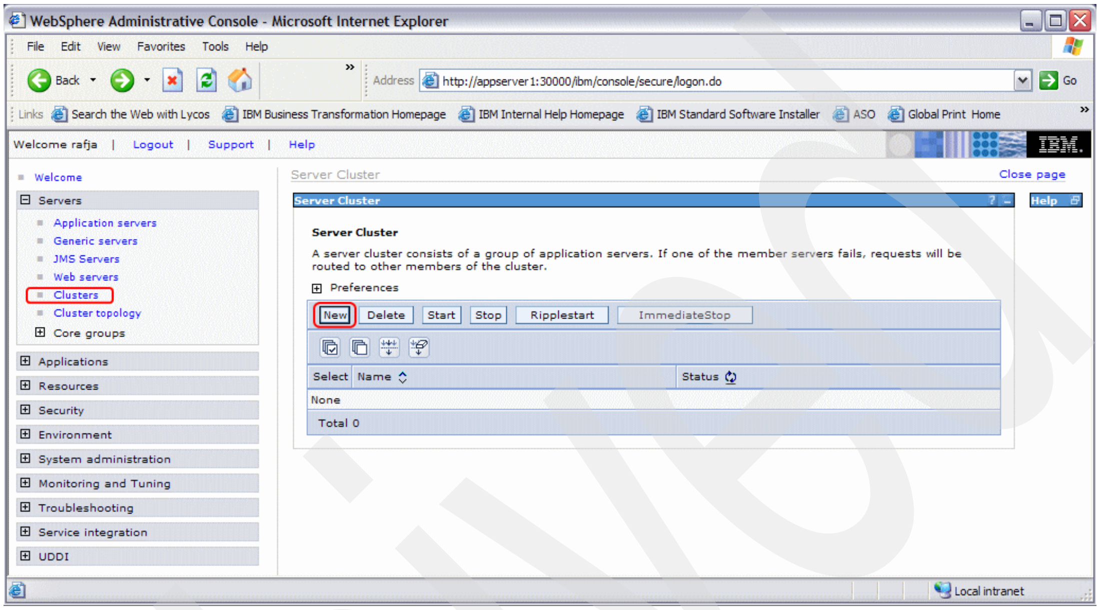
Table 5-8 lists the parameters that are used to create a WebSphere Application Server cluster and cluster members.

Table 5-8 WebSphere Application Server profile parameters

Host name	Cluster name	Cluster member name	Node name
APPSERVER1	ITSOCluster	Member1	APPSERVER1_AppSvr1
APPSERVER1	ITSOCluster	Member2	APPSERVER1_AppSvr1
APPSERVER2	ITSOCluster	Member3	APPSERVER2_AppSvr2
APPSERVER2	ITSOCluster	Member4	APPSERVER2_AppSvr2

To create a WebSphere Application Server cluster, use these steps:

1. Open the deployment manager WebSphere Administrative Console.
2. Expand **Servers** and click **Clusters**. In the Server Cluster pane on the right, click **New** (see Figure 5-67).


WebSphere Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://appserver1:30000/ibm/console/secure/logon.do
Go
Links Search the Web with Lycos IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer ASO Global Print Home
Welcome rafja | Logout | Support | Help
IBM.
Welcome
Servers
Application servers
Generic servers
JMS Servers
Web servers
Clusters
Cluster topology
Core groups
Applications
Resources
Security
Environment
System administration
Monitoring and Tuning
Troubleshooting
Service integration
UDDI
Server Cluster
Close page
Help
New
Delete
Start
Stop
Ripplestart
ImmediateStop
Local intranet

Server Cluster

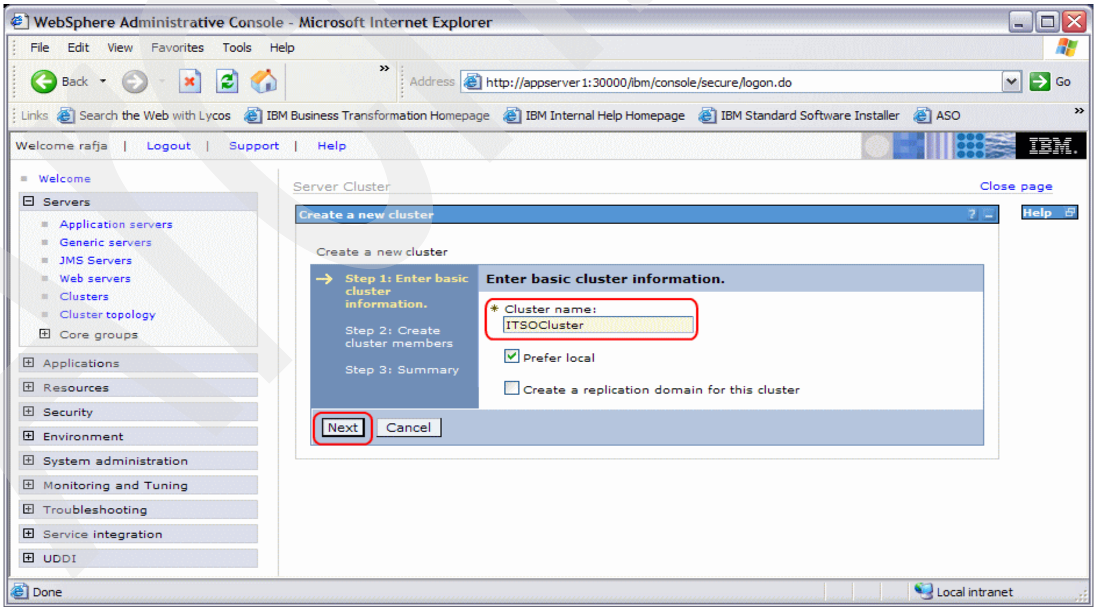
A server cluster consists of a group of application servers. If one of the member servers fails, requests will be routed to other members of the cluster.

⊞ Preferences

Select	Name	Status
None		
Total 0		

Figure 5-67 WebSphere Application Server clusters

3. In the Create a new cluster pane (Figure 5-68), in the Cluster name field, type a name for the new cluster. In this scenario, we typed `ITSOCluster` in this field. Click **Next**.


WebSphere Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://appserver1:30000/ibm/console/secure/logon.do
Go
Links Search the Web with Lycos IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer ASO
Welcome rafja | Logout | Support | Help
IBM.
Welcome
Servers
Application servers
Generic servers
JMS Servers
Web servers
Clusters
Cluster topology
Core groups
Applications
Resources
Security
Environment
System administration
Monitoring and Tuning
Troubleshooting
Service integration
UDDI
Server Cluster
Close page
Create a new cluster
Help
Step 1: Enter basic cluster information.
Step 2: Create cluster members
Step 3: Summary
Next
Cancel
Done
Local intranet

Create a new cluster

Enter basic cluster information.

* Cluster name: ITSOCluster

☑ Prefer local

□ Create a replication domain for this cluster

Figure 5-68 Creating a new cluster

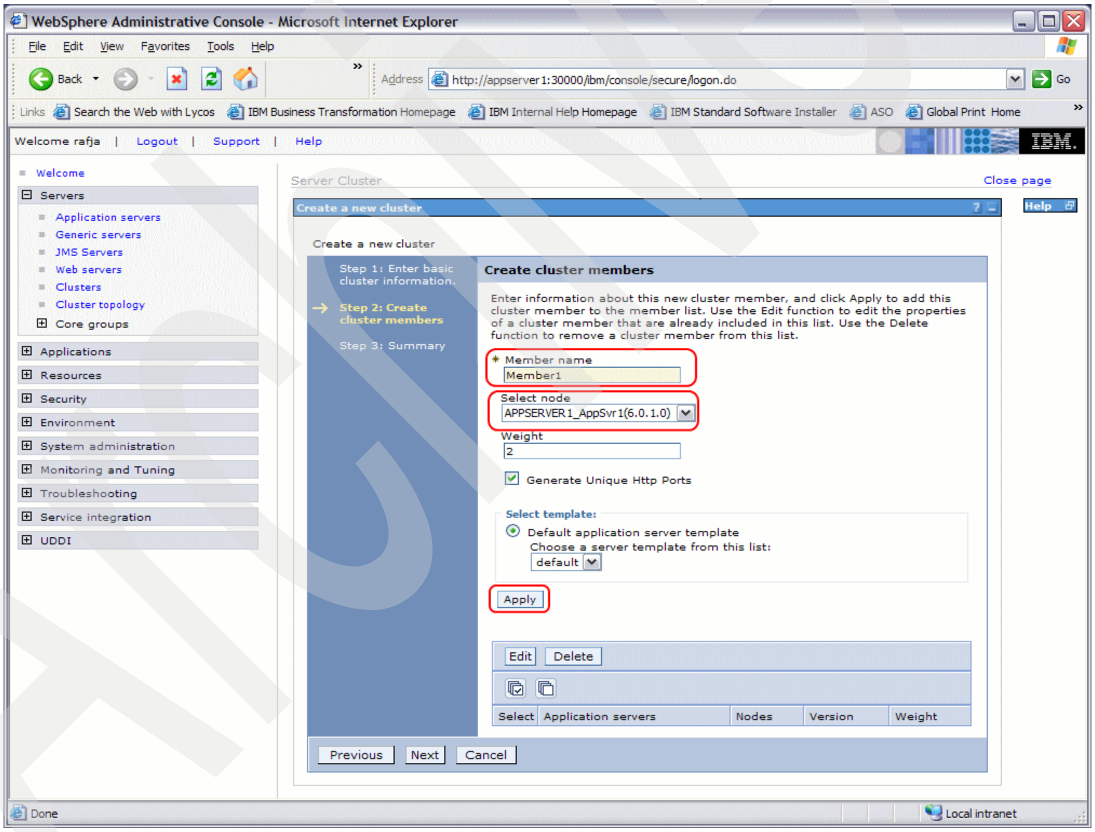
4. In the next panel (Figure 5-69), you enter information about a new cluster. A *cluster member* is an application server that processes user requests. In this example, four cluster members are created, two on each of the iSeries servers.

   a. In the Member name field, specify the cluster member name. You can enter any name here. This name is assigned to the application server. Later, the name of the job, representing this cluster member, that runs in the QWAS6 subsystem will have the name that is assigned in this step. In this example, we typed `Member1` for the first cluster member.

   b. For Select node, specify the node on which the cluster member will be created. The first two cluster members are created on the APPSERVER1 iSeries. In this example, we selected **APPSERVER1_AppSvr1**.

   c. For Weight, specify a value which represents the performance capabilities of the cluster member. You can modify this value depending on the commercial processing workload (CPW) of the iSeries. Assign a higher weight value on a higher CPW iSeries and assign a lower weight value on a lower CPW iSeries. In this scenario, we accepted the default value.

   d. Click **Apply** to create the cluster member.



Enter information about this new cluster member, and click Apply to add this cluster member to the member list. Use the Edit function to edit the properties of a cluster member that are already included in this list. Use the Delete function to remove a cluster member from this list.

* Member name: Member1

Select node: APPSERVER1_AppSvr1(6.0.1.0)

Weight: 2

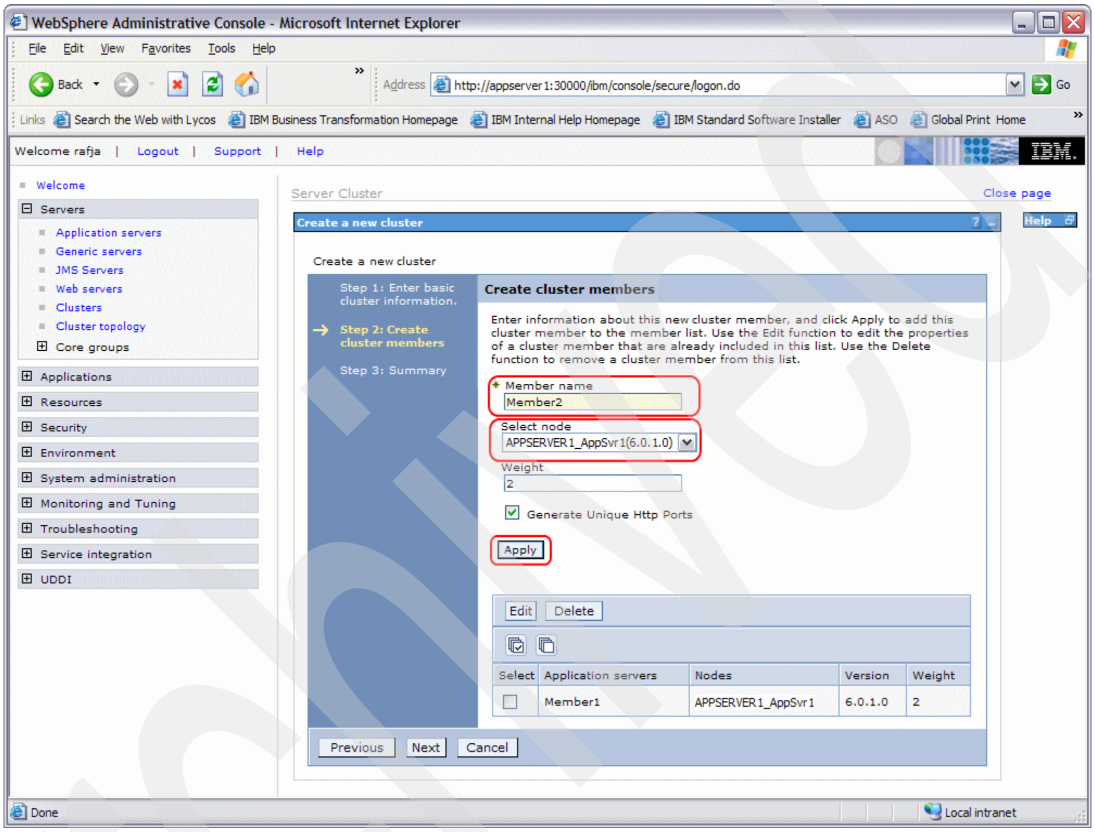
☑ Generate Unique Http Ports

Select template:
● Default application server template
Choose a server template from this list: default

Figure 5-69 Creating the first cluster member

5. After you add the first cluster member to the cluster, the window refreshes and you can add more cluster members. Continue to add another cluster member. For this cluster member, specify the following values as shown in Figure 5-70:

   a. For Member name, type `Member2`.
   b. For Select node, select **APPSERVER1_AppSvr1**.
   c. For Weight, keep the default value.
   d. Click **Apply**.


WebSphere Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://appserver1:30000/ibm/console/secure/logon.do
Go
Links Search the Web with Lycos IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer ASO Global Print Home
Welcome rafja | Logout | Support | Help
IBM.
Welcome
Servers
Application servers
Generic servers
JMS Servers
Web servers
Clusters
Cluster topology
Core groups
Applications
Resources
Security
Environment
System administration
Monitoring and Tuning
Troubleshooting
Service integration
UDDI
Server Cluster
Close page
Help
Step 1: Enter basic cluster information.
Step 2: Create cluster members
Step 3: Summary
Apply
Edit
Delete
Previous
Next
Cancel
Done
Local intranet

Create a new cluster

Create a new cluster

Create cluster members

Enter information about this new cluster member, and click Apply to add this cluster member to the member list. Use the Edit function to edit the properties of a cluster member that are already included in this list. Use the Delete function to remove a cluster member from this list.

* Member name: Member2

Select node: APPSERVER1_AppSvr1(6.0.1.0)

Weight: 2

☑ Generate Unique Http Ports

Select	Application servers	Nodes	Version	Weight
□	Member1	APPSERVER1_AppSvr1	6.0.1.0	2

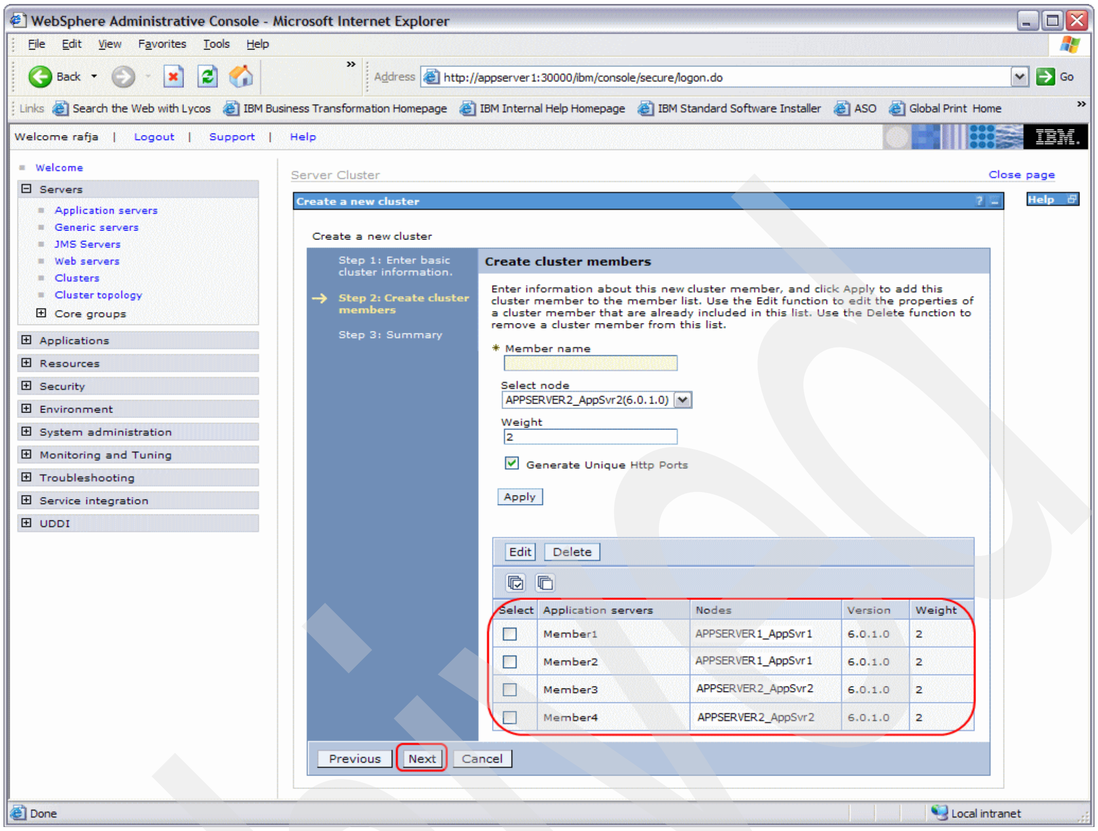
Figure 5-70 Creating the second cluster member

6. After the second cluster member is added to the cluster, the window refreshes and you can add more cluster members. For this scenario, we require two cluster members on the first node (which we just configured) and two cluster members on the second node.

   Continue to use the deployment manager WebSphere Administrative Console and create two cluster members for node APPSERVER2_AppSvr2. To create the next two cluster members repeat step 4 on page 150. Refer to Table 5-8 on page 148 for the names of these new cluster members. In this scenario, specify the following values for each of the remaining members:

   a. For member name, type `Member3` for the third member and `Member4` for the fourth member.
   b. For Select node, choose **APPSERVER2_AppSvr2**
   c. For Weight, accept the default value.
   d. Click **Apply**.

7. After you create all four cluster members, you see the members in the Create cluster members window as shown in Figure 5-71. Click **Next**.


WebSphere Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://appserver1:30000/ibm/console/secure/logon.do
Go
Links Search the Web with Lycos IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer ASO Global Print Home
Welcome rafja | Logout | Support | Help
IBM.
Welcome
Servers
Application servers
Generic servers
JMS Servers
Web servers
Clusters
Cluster topology
Core groups
Applications
Resources
Security
Environment
System administration
Monitoring and Tuning
Troubleshooting
Service integration
UDDI
Server Cluster
Close page
Help
Done
Local intranet

Create a new cluster

Create a new cluster

Step 1: Enter basic cluster information.

→ Step 2: Create cluster members

Step 3: Summary

Create cluster members

Enter information about this new cluster member, and click Apply to add this cluster member to the member list. Use the Edit function to edit the properties of a cluster member that are already included in this list. Use the Delete function to remove a cluster member from this list.

* Member name

Select node
APPSERVER2_AppSvr2(6.0.1.0)

Weight
2

☑ Generate Unique Http Ports

Apply

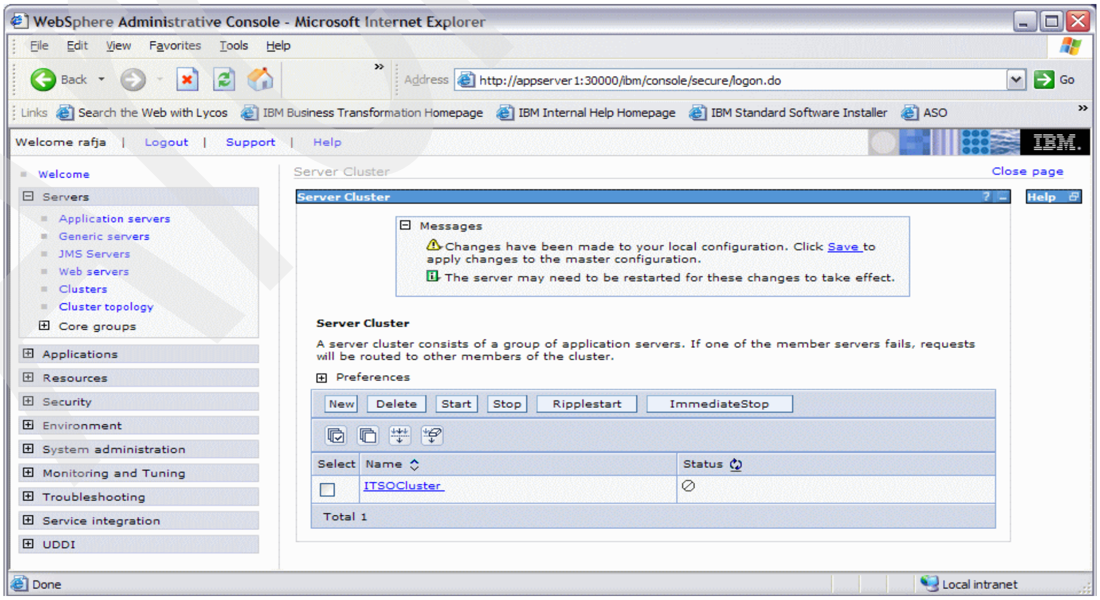
Edit Delete

Select	Application servers	Nodes	Version	Weight
□	Member1	APPSERVER1_AppSvr1	6.0.1.0	2
□	Member2	APPSERVER1_AppSvr1	6.0.1.0	2
□	Member3	APPSERVER2_AppSvr2	6.0.1.0	2
□	Member4	APPSERVER2_AppSvr2	6.0.1.0	2

Previous Next Cancel

Figure 5-71 Cluster members list

8. In the Summary pane, click **Finish** to add all cluster members to the cluster.
9. After you add all cluster members, you see a window like the one in Figure 5-72. To save this newly created configuration, click **Save**.


WebSphere Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://appserver1:30000/ibm/console/secure/logon.do
Go
Links Search the Web with Lycos IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer ASO
Welcome rafja | Logout | Support | Help
IBM.
Welcome
Servers
Application servers
Generic servers
JMS Servers
Web servers
Clusters
Cluster topology
Core groups
Applications
Resources
Security
Environment
System administration
Monitoring and Tuning
Troubleshooting
Service integration
UDDI
Server Cluster
Close page
Help
Done
Local intranet

Server Cluster

Messages

⚠ Changes have been made to your local configuration. Click Save to apply changes to the master configuration.

ℹ The server may need to be restarted for these changes to take effect.

Server Cluster

A server cluster consists of a group of application servers. If one of the member servers fails, requests will be routed to other members of the cluster.

⊞ Preferences

New Delete Start Stop Ripplestart ImmediateStop

Select	Name	Status
□	ITSOCluster	⊘
Total 1		

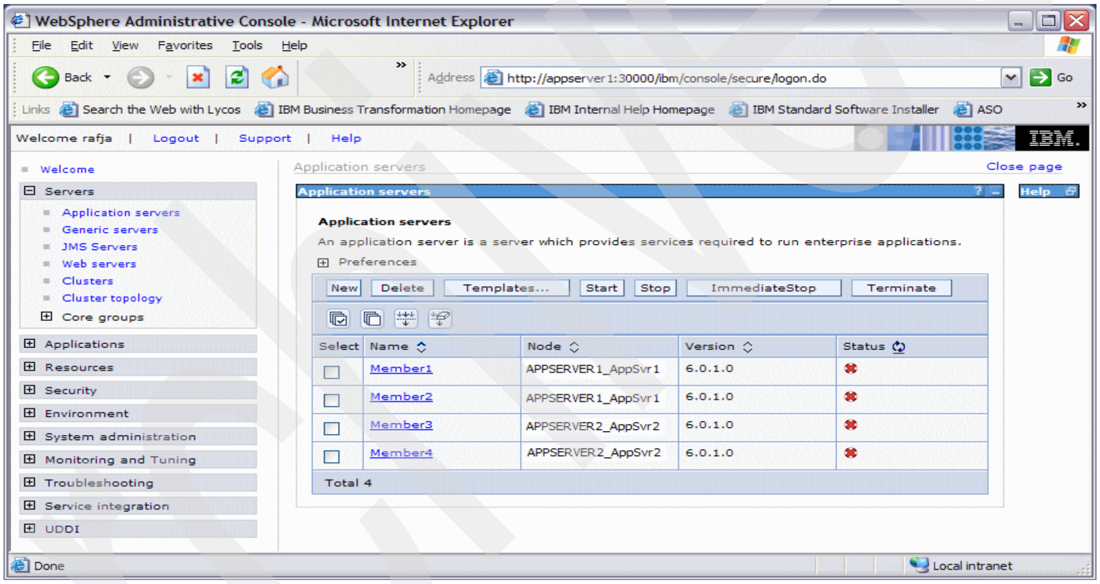
Figure 5-72 Created cluster

Virtual hosts update

The concept of *virtual hosts*, in terms of Web serving, refers to the practice of maintaining more than one domain in a single server. Domains are differentiated by their host name or IP address. Client requests are routed to the correct domain by IP address or by the host name contained in the URL header. Virtual hosts control which Web servers can access which application servers.

To set up virtual hosts, you must check the TCP/IP ports that each of the cluster member listens to. You can only check these ports after you create the cluster members. During creation of the cluster members, HTTP ports are automatically assigned to each cluster member.

To see which TCP/IP ports are assigned to the cluster members, return to the deployment manager WebSphere Administrative Console. Expand **Servers** and click **Application servers**. The resulting panel looks like the example in Figure 5-73. This panel shows four application servers that are defined. These four application servers relate to the four cluster members that were created earlier.



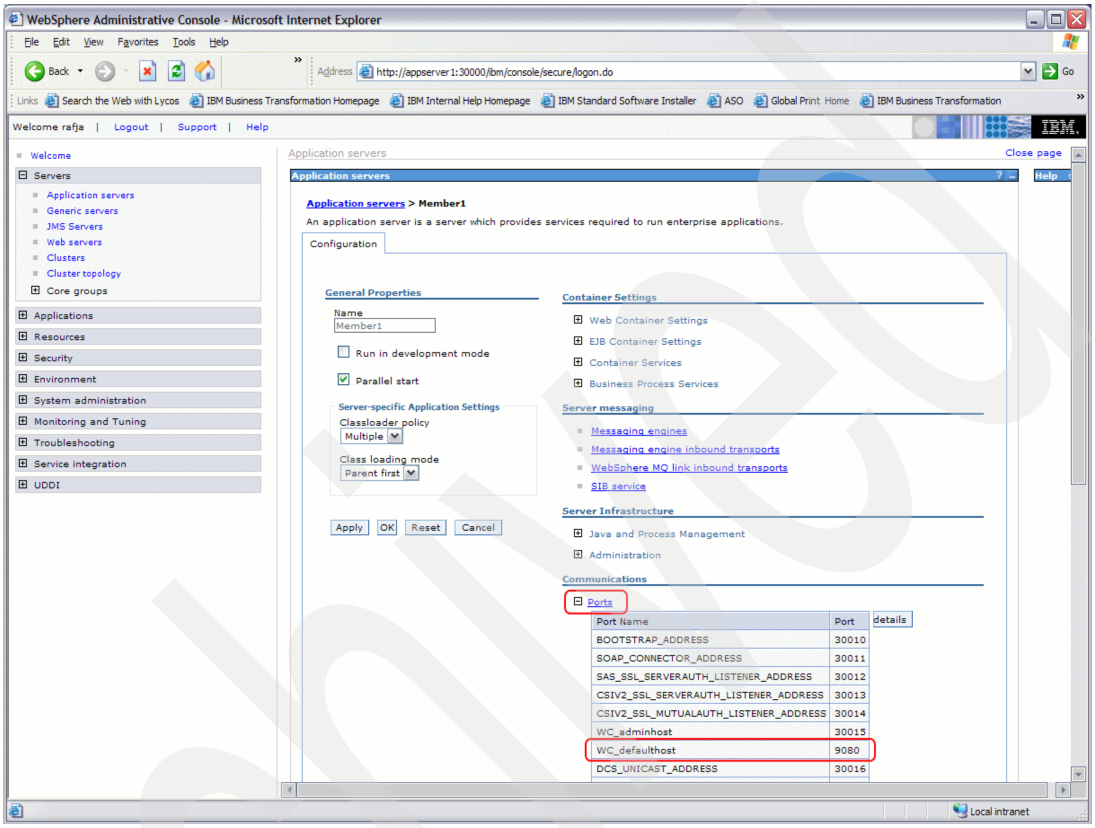
Application servers

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

Select	Name	Node	Version	Status
□	Member1	APPSERVER1_AppSvr1	6.0.1.0	✖
□	Member2	APPSERVER1_AppSvr1	6.0.1.0	✖
□	Member3	APPSERVER2_AppSvr2	6.0.1.0	✖
□	Member4	APPSERVER2_AppSvr2	6.0.1.0	✖
Total 4				

Figure 5-73 Application servers list

When you click any of the application servers, you see a window similar to the one shown in Figure 5-74. Under Communications, either expand **Ports** or click **Ports**. In either case, you see additional port information. Record the port that corresponds to *WC_defaulthost*, which relates to the Web container HTTP port. You need this port to correctly set up the virtual hosts information.



Application servers > Member1

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

General Properties

Name: Member1

☐ Run in development mode

☑ Parallel start

Server-specific Application Settings

Classloader policy: Multiple

Class loading mode: Parent first

Port Name	Port
BOOTSTRAP_ADDRESS	30010
SOAP_CONNECTOR_ADDRESS	30011
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS	30012
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS	30013
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS	30014
WC_adminhost	30015
WC_defaulthost	9080
DCS_UNICAST_ADDRESS	30016

Figure 5-74 Application server details

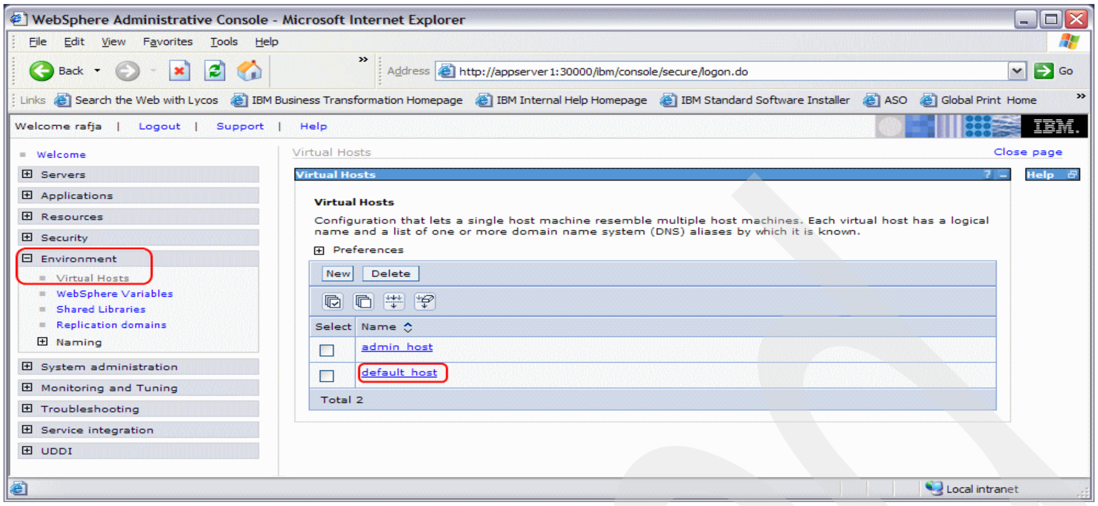
Continue to find the WC_defaulthost port for each application server that has been defined in this scenario. Record this information in a table similar to the one shown in Table 5-9, which identifies the cluster member, the host name, and the WC_defaulthost value.

Table 5-9 Cluster members' port setup

Cluster member name	Host name	*WC_defaulthost* value
Member1	APPSERVER1	9080
Member2	APPSERVER1	9081
Member3	APPSERVER2	9081
Member4	APPSERVER2	9082

Based on the information in Table 5-9, you can update the virtual hosts.

1. Virtual hosts information is found in the deployment manager WebSphere Administrative Console. Expand **Environment** and then click **Virtual Hosts**.
2. The resulting window is shown in Figure 5-75. Two virtual hosts are defined by default. The first (admin_host) is dedicated to run administration console. The second one (default_host) is dedicated to business applications. Click **default_host**.


WebSphere Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://appserver1:30000/ibm/console/secure/logon.do
Go
Links Search the Web with Lycos IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer ASO Global Print Home
Welcome rafja | Logout | Support | Help
IBM.
Welcome
Servers
Applications
Resources
Security
Environment
Virtual Hosts
WebSphere Variables
Shared Libraries
Replication domains
Naming
System administration
Monitoring and Tuning
Troubleshooting
Service integration
UDDI
Virtual Hosts
Close page
Help
Preferences
New
Delete
Local intranet

Virtual Hosts

Configuration that lets a single host machine resemble multiple host machines. Each virtual host has a logical name and a list of one or more domain name system (DNS) aliases by which it is known.

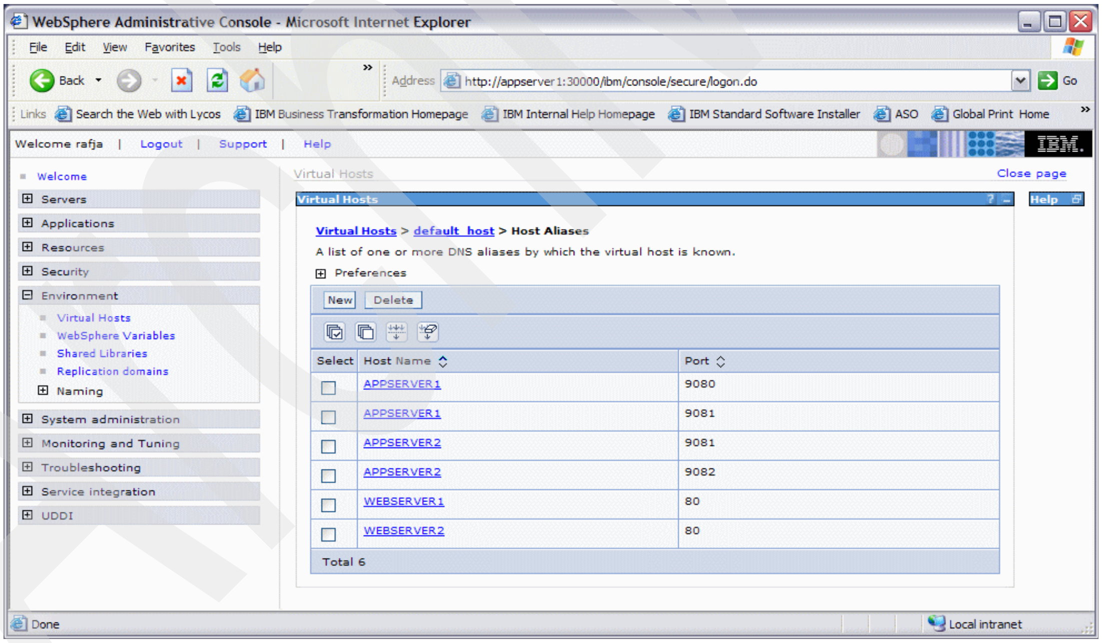
Select	Name
□	admin_host
□	default_host

Total 2

Figure 5-75 Virtual hosts definition

3. Under Additional Properties, click **Host Aliases**.

4. Update the host alias definitions according to Table 5-9 (see Figure 5-76). The last two entries shown in Figure 5-76 relate to the Web server systems host names: WEBSERVER1 and WEBSERVER2. Both of these Web servers listen on default HTTP port 80.


WebSphere Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://appserver1:30000/ibm/console/secure/logon.do
Go
Links Search the Web with Lycos IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer ASO Global Print Home
Welcome rafja | Logout | Support | Help
IBM.
Welcome
Servers
Applications
Resources
Security
Environment
Virtual Hosts
WebSphere Variables
Shared Libraries
Replication domains
Naming
System administration
Monitoring and Tuning
Troubleshooting
Service integration
UDDI
Virtual Hosts
Close page
Help
Preferences
New
Delete
Done
Local intranet

Virtual Hosts > default_host > Host Aliases

A list of one or more DNS aliases by which the virtual host is known.

Select	Host Name	Port
□	APPSERVER1	9080
□	APPSERVER1	9081
□	APPSERVER2	9081
□	APPSERVER2	9082
□	WEBSERVER1	80
□	WEBSERVER2	80

Total 6

Figure 5-76 Host aliases

Now that all servers are defined, we can start configuring the Web server definitions.

Creating Web server definitions

To establish a connection between the HTTP Server instance and WebSphere Application Server profile, you must create a Web server definition on the application server side. In this scenario, there are two Web server instances, which means that we need to create two Web server definitions, one for each HTTP server instance.

Before you create a Web server definition, you need the parameters required to run the configuration script. Table 5-10 lists the parameters that are used to run the configuration script.

Table 5-10 Web server definition script parameters

WebSphere profile name	Web server instance name	Web server instance port	Web server definition name
DmgrSvr	WebSvr1	80	IHS_WEBSERVER1_WebSvr1
DmgrSvr	WebSvr2	80	IHS_WEBSERVER2_WebSvr2

In Table 5-10, the first parameter (DmgrSvr) was created in “Creating deployment manager profile” on page 141. The second and third parameters (WebSvr1, WebSvr2, and 80) were created in 5.5.2, “Creating HTTP Server instances” on page 140. The fourth parameter is the Web server definition name. This name is constructed according to the following template:

webserverType_hostName_webserverInstanceName

Note the following explanation:

- *webserverType*: This is the type of Web server. A value of IHS or DOMINO is accepted. For this scenario the type is IHS (IBM HTTP Server).
- *hostName*: This is the iSeries host name for the Web server.
- *webserverInstanceName*: This is the HTTP Server instance name.

Based on the parameters shown in Table 5-10, use the following steps to create a Web server definition:

1. Log on to the APPSERVER1 host, where the deployment manager is created.
2. On the OS/400 command line, type the `STRQSH` command.
3. On the Qshell command line, type the following command:

   ```
   cd /QIBM/ProdData/WebSphere/AppServer/V6/Base/bin
   ```

4. Run the configureOs400WebServerDefinition script based on the parameters in the first row in Table 5-10. For this scenario, the command is:

   ```
   configureOs400WebServerDefinition -profileName DmgrSvr -webserver.instance.name WebSvr1
   -webserver.port 80 -webserver.name IHS_WEBSERVER1_WebSvr1
   ```

   After the script is invoked, you see messages similar to those in Example 5-4 in the Qshell session.

   Example 5-4 Web server definition configuration output

   ```
   > configureOs400WebServerDefinition -profileName DmgrSvr -webserver.instance.name
   WebSvr1
   -webserver.port 80 -webserver.name IHS_WEBSERVER1_WebSvr1
     ...
   Input parameters:
       Web server name             - IHS_WEBSERVER1_WebSvr1
       Web server type           - IHS
       Web server install location - WebSvr1
       Web server config location  - /www/WebSvr1/conf/httpd.conf
       Web server port           - 80
       Web server admin port      - 2001
       Map Applications          - MAP_ALL
       Plugin install location     -
   /QIBM/UserData/WebSphere/AppServer/V6/Base/profiles/DmgrSvr
   ```

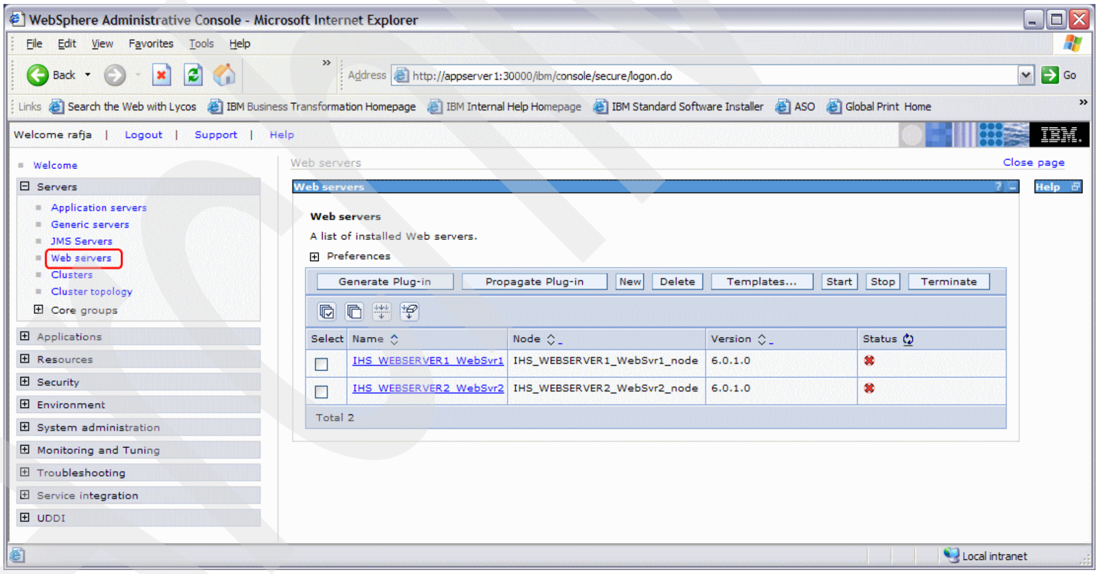
```
   Web server node type       - unmanaged
   Web server node name       - IHS_WEBSERVER1_WebSvr1_node
   Web server host name        - WEBSEVER1
   Web server operating system - os400

  Creating the unmanaged node IHS_WEBSERVER1_WebSvr1_node .
  Unmanged node IHS_WEBSERVER1_WebSvr1_node is created.

  Creating the web server definition for IHS_WEBSERVER1_WebSvr1.
Target mapping is updated for the application query.
  Start saving the configuration.
  Configuration save is complete.
  $
```

5. Run the configureOs400WebServerDefinition script again, based on the second row in Table 5-10. For this scenario, the command is:

   ```
   configureOs400WebServerDefinition -profileName DmgrSvr -webserver.instance.name WebSvr2
   -webserver.port 80 -webserver.name IHS_WEBSERVER2_WebSvr2
   ```

6. In a Web browser, open the deployment manager Admin Console.

7. Expand **Servers** and click **Web servers**. You should see two Web server definitions in the right pane as shown in Figure 5-77. The first entry represents the Web server located on the WEBSERVER1 iSeries, and the second entry represents the Web server located on the WEBSERVER2 host.


WebSphere Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://appserver1:30000/ibm/console/secure/logon.do
Go
Links Search the Web with Lycos IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer ASO Global Print Home
Welcome rafja | Logout | Support | Help
IBM.
Welcome
Servers
Application servers
Generic servers
JMS Servers
Web servers
Clusters
Cluster topology
Core groups
Applications
Resources
Security
Environment
System administration
Monitoring and Tuning
Troubleshooting
Service integration
UDDI
Web servers
Close page
Help
Local intranet

Web servers

A list of installed Web servers.

⊞ Preferences

Generate Plug-in | Propagate Plug-in | New | Delete | Templates... | Start | Stop | Terminate

Select	Name	Node	Version	Status
□	IHS_WEBSERVER1_WebSvr1	IHS_WEBSERVER1_WebSvr1_node	6.0.1.0	✖
□	IHS_WEBSERVER2_WebSvr2	IHS_WEBSERVER2_WebSvr2_node	6.0.1.0	✖

Total 2

Figure 5-77 Web server definitions

Now, it’s time to install the application. We use Trade6 for our example. You may install your own application.

Installing Trade 6

If you want to use the sample application to test your configuration, see Appendix C, “Installing the Trade6 application” on page 277, for more information.

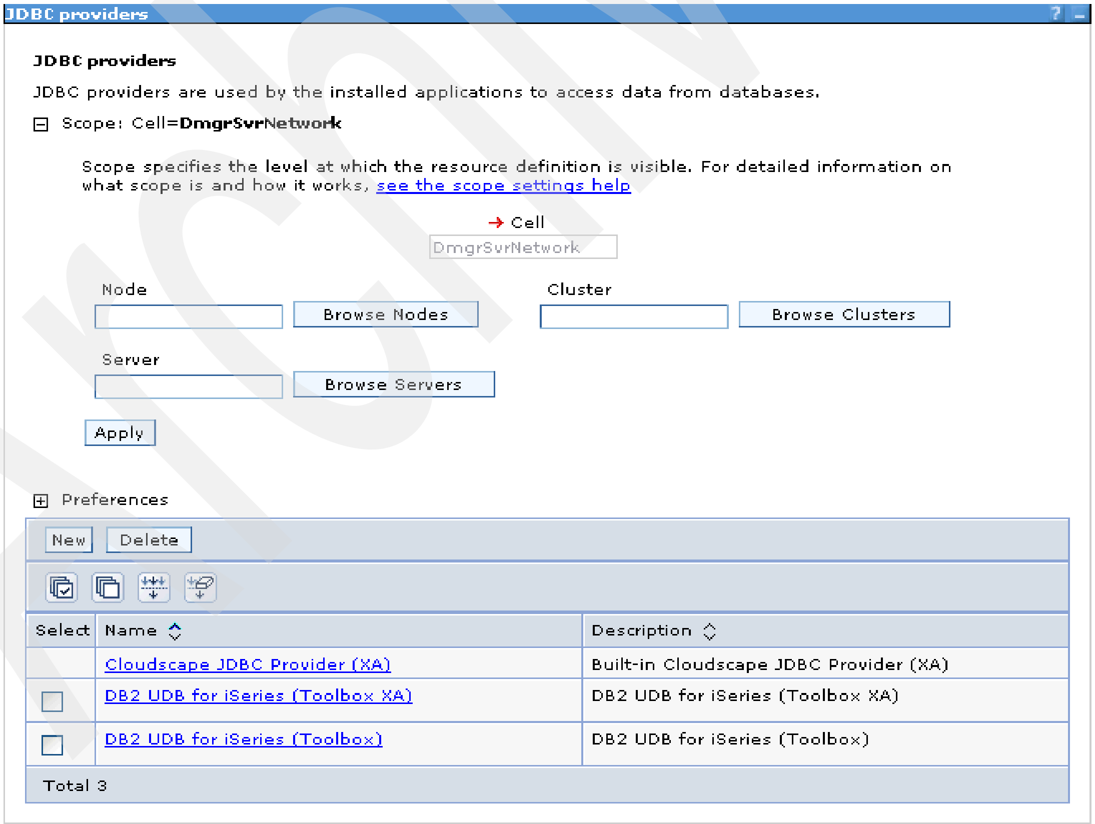
> **Important:** If you plan to use your own application, you must review all sections for the Trade6 application. They contain information that pertains to any application that you install into a cluster.

WebSphere data source changes for the Trade6 application

Now that the TRADE6 database is restored on the primary cluster node IASP, make sure that the TRADE6 enterprise application can access the data on the Trade IASP in the XSM configuration.

We specified to use the Toolbox XA driver for our Trade6 installation. This causes two Java Database Connectivity (JDBC) Toolbox providers to be created: one for the Trade6 database application access and the other for the message engine data store for message persistence. Each one of the provider's data source must be modified to correctly access the information in our database protected by XSM. Also additional performance and HA custom properties must be configured to optimize the failover conditions.

1. Modify the JDBC providers for Trade6 and XSM.
   a. In the administrative console, expand **Resources** and click **JDBC providers**.
   b. In the JDBC providers panel (Figure 5-78), make sure that the scope is set at the cell level. Then click the **DB2 UDB for iSeries (Toolbox XA)** provider that was created by the Trade6 installation script. This provider has a data source associated with it which defines access to the Trade6 database.



JDBC providers

JDBC providers are used by the installed applications to access data from databases.

⊟ Scope: Cell=DmgrSvrNetwork

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help

→ Cell: DmgrSvrNetwork

Node: ______ Browse Nodes

Cluster: ______ Browse Clusters

Server: ______ Browse Servers

Apply

⊞ Preferences

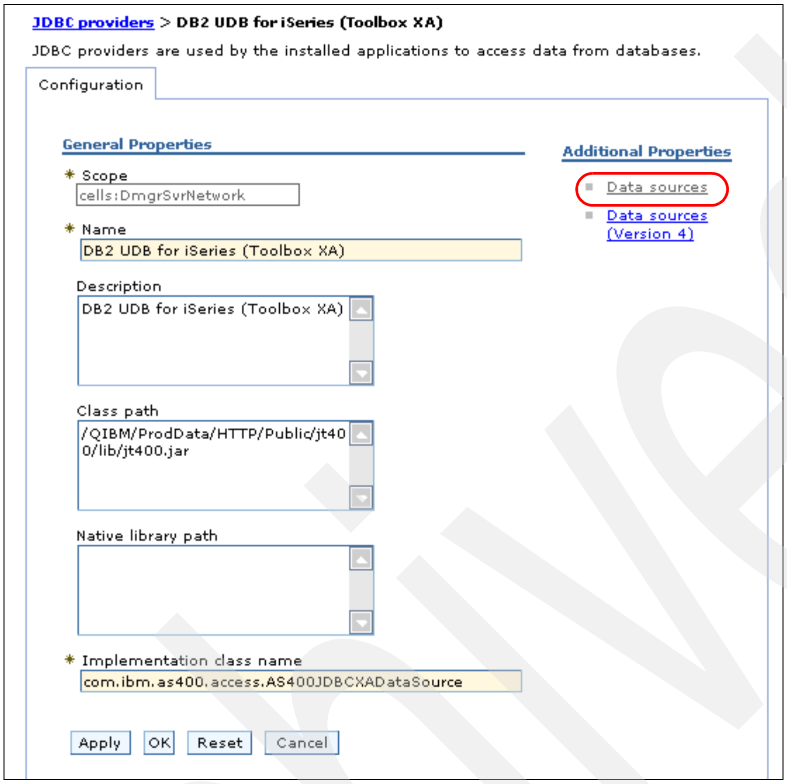
New Delete

Select	Name ◇	Description ◇
	Cloudscape JDBC Provider (XA)	Built-in Cloudscape JDBC Provider (XA)
□	DB2 UDB for iSeries (Toolbox XA)	DB2 UDB for iSeries (Toolbox XA)
□	DB2 UDB for iSeries (Toolbox)	DB2 UDB for iSeries (Toolbox)

Total 3

Figure 5-78 JDBC resources that need to be modified

c. Verify the data source for Trade6 database access. On the JDBC providers → DB2 UDB for iSeries (Toolbox XA) panel (Figure 5-79), under Additional Properties, click **Data Sources**. The data source describes the characteristics for accessing data with the given provider. This is where you verify the XSM information.



JDBC providers > DB2 UDB for iSeries (Toolbox XA)

JDBC providers are used by the installed applications to access data from databases.

Configuration

General Properties

* Scope: cells:DmgrSvrNetwork

* Name: DB2 UDB for iSeries (Toolbox XA)

Description: DB2 UDB for iSeries (Toolbox XA)

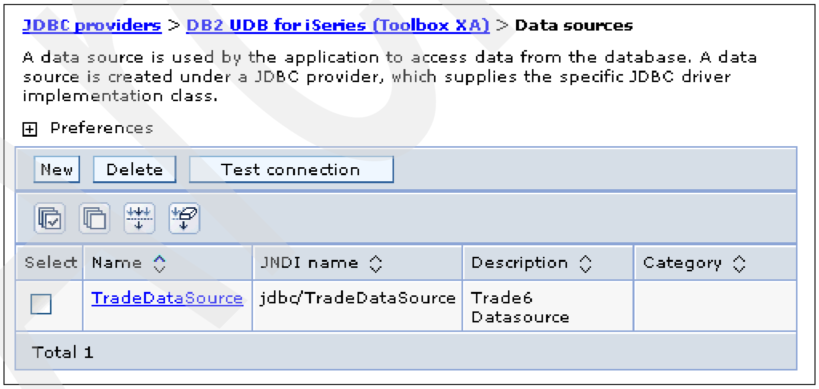
Class path: /QIBM/ProdData/HTTP/Public/jt400/lib/jt400.jar

Native library path:

* Implementation class name: com.ibm.as400.access.AS400JDBCXADataSource

Figure 5-79 JDBC provider information

2. As shown in Figure 5-80, click **TradeDataSource**.



JDBC providers > DB2 UDB for iSeries (Toolbox XA) > Data sources

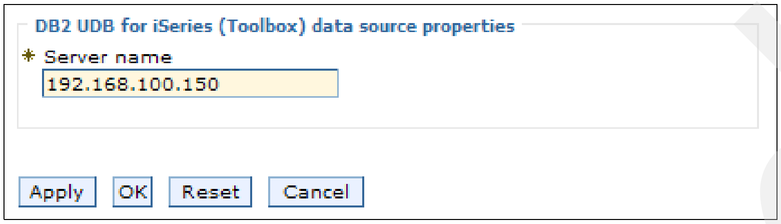
A data source is used by the application to access data from the database. A data source is created under a JDBC provider, which supplies the specific JDBC driver implementation class.

Select	Name	JNDI name	Description	Category
□	TradeDataSource	jdbc/TradeDataSource	Trade6 Datasource	

Total 1

Figure 5-80 Selecting to verify TradeDataSource

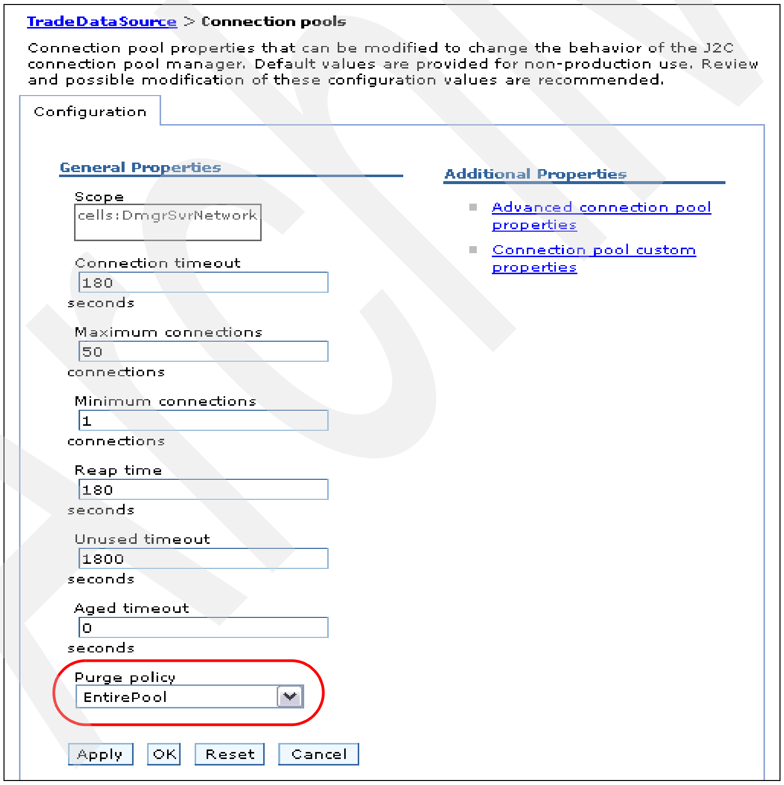
3. View the general properties for the data source and make sure that the server name specified at the bottom of the properties form has the XSM takeover IP address (see Figure 5-81). This allows the data source to access data via the XSM IP address that we set up earlier. With this change, when the IASP switchover happens, we don't need to modify WebSphere settings.


DB2 UDB for iSeries (Toolbox) data source properties

* Server name
192.168.100.150

Figure 5-81 Pointing to the takeover IP address

4. Modify the connection pool properties for the data source (see Figure 5-82). By default, the purge policy for a bad connection in the pool is set to connection only. We need to set this value to have the entire connection pool recreated after XSM switchover.
   a. From the data source general properties pane, under Additional Properties, click **Connection pool properties**.
   b. For the Purge policy parameter, select **EntirePool**.
   c. Click **Apply**.


TradeDataSource > Connection pools

Connection pool properties that can be modified to change the behavior of the J2C connection pool manager. Default values are provided for non-production use. Review and possible modification of these configuration values are recommended.

Configuration

General Properties

Scope
cells:DmgrSvrNetwork

Connection timeout
180
seconds

Maximum connections
50
connections

Minimum connections
1
connections

Reap time
180
seconds

Unused timeout
1800
seconds

Aged timeout
0
seconds

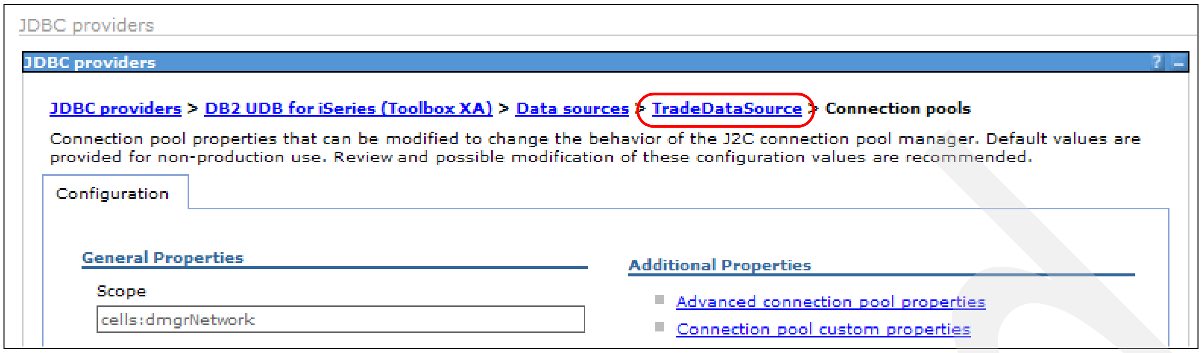
Purge policy
EntirePool

Additional Properties

- Advanced connection pool properties
- Connection pool custom properties

Figure 5-82 Connection pool purge policy

5. Click the **TradeDataSource** link at the top of your Web browser page (see Figure 5-83).



JDBC providers

JDBC providers

JDBC providers > DB2 UDB for iSeries (Toolbox XA) > Data sources > TradeDataSource > **Connection pools**

Connection pool properties that can be modified to change the behavior of the J2C connection pool manager. Default values are provided for non-production use. Review and possible modification of these configuration values are recommended.

Configuration

General Properties

Scope

cells:dmgrNetwork

Additional Properties

- Advanced connection pool properties
- Connection pool custom properties

Figure 5-83 Navigating to the data source properties page

6. You return to the Data source properties page. Modify the custom properties for the data source to allow access to our IASP Trade where our database resides. Also, failover performance settings are added.

   a. Under Additional properties, click **Custom properties**.

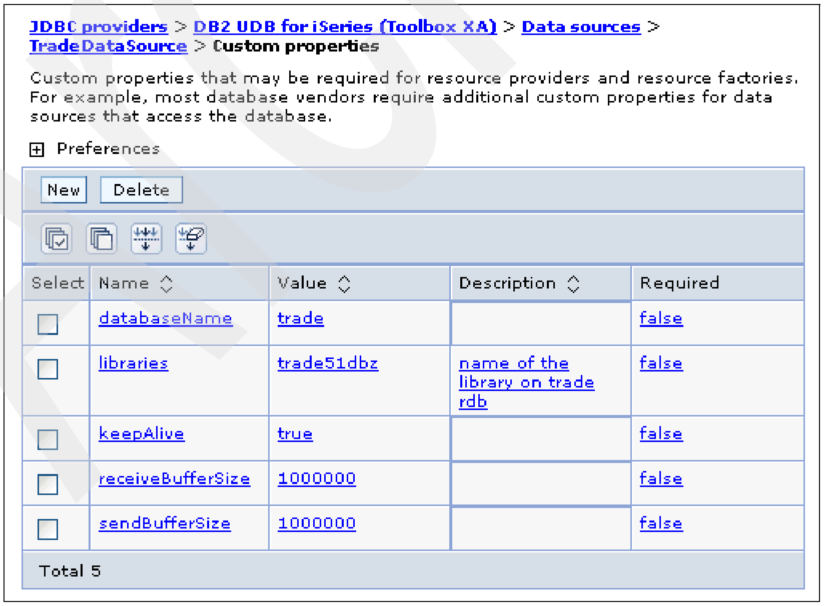
   b. Add the properties as shown in Figure 5-84. The custom properties apply only to Toolbox data sources.

      i. The databaseName property should already be there; it’s the result of running the script for Trade6.

      ii. A databaseName property value is required to access databases that reside in an IASP. Set the value to the name that was given to the independent disk pool. In our case, ‘trade’ was the name assigned to our switchable disk pool.

      iii. Set the keepAlive property to a value of `true`, which is critical to help ensure high-availability characteristics in the event of database failover processing.

   c. Save your changes.



JDBC providers > DB2 UDB for iSeries (Toolbox XA) > Data sources > TradeDataSource > **Custom properties**

Custom properties that may be required for resource providers and resource factories. For example, most database vendors require additional custom properties for data sources that access the database.

⊞ Preferences

Select	Name	Value	Description	Required
□	databaseName	trade		false
□	libraries	trade51dbz	name of the library on trade rdb	false
□	keepAlive	true		false
□	receiveBufferSize	1000000		false
□	sendBufferSize	1000000		false

Total 5

Figure 5-84 Custom properties for Trade6 and XSM

WebSphere transaction log changes

To make a transaction log highly available, we place it on the IASP, protected with geographic mirroring (see 5.3, “Data considerations for high availability” on page 99). We use the same IASP for the transaction log as for the application database. This configuration option requires the use of the QNTC file system.

Making the WebSphere transaction logs highly available is needed only for applications that use the WebSphere Transaction Manager to coordinate XA transactions. The TRADE6 application uses XA data sources, so the WebSphere transaction logs need to be made highly available.

In this section, you learn how to configure the QNTC file system and the WebSphere Application Server cluster to support the remote transaction log (transaction log is replicated by geographic mirroring). We perform the following tasks:

1. Configure the QNTC file system.
2. Create the directory structure for each application server to hold the transaction log.
3. Configure each WebSphere server to point to the remote transaction log file.

Configuring the transaction log directories in the QNTC file system

As mentioned earlier, we place the transaction log on the IASP that we configured for the application database. This ensures that both the transaction logs and the database will become available at the same time after disk pool failover processing is complete. Also, we recommend that database access paths be journaled so that all database files will be available to the application for SQL operations immediately after the takeover IP address comes active at the end of the failover processing. We use the iSeries cluster takeover IP address to access this IASP (see “Setting the takeover IP address” on page 125) and the QNTC file system.

To add support for the remote IASP, we create a new directory under /QNTC. Perform these steps on *each* iSeries that has WebSphere servers:

1. Start QShell:

   ```
   strqsh
   ```

2. Run the following command, replacing the IP address with the takeover IP address that you’ve used for the iSeries cluster:

   ```
   mkdir /QNTC/192.168.100.150
   ```

3. It takes a while to connect to the remote system. After the previous command completes, change to the remote file system.

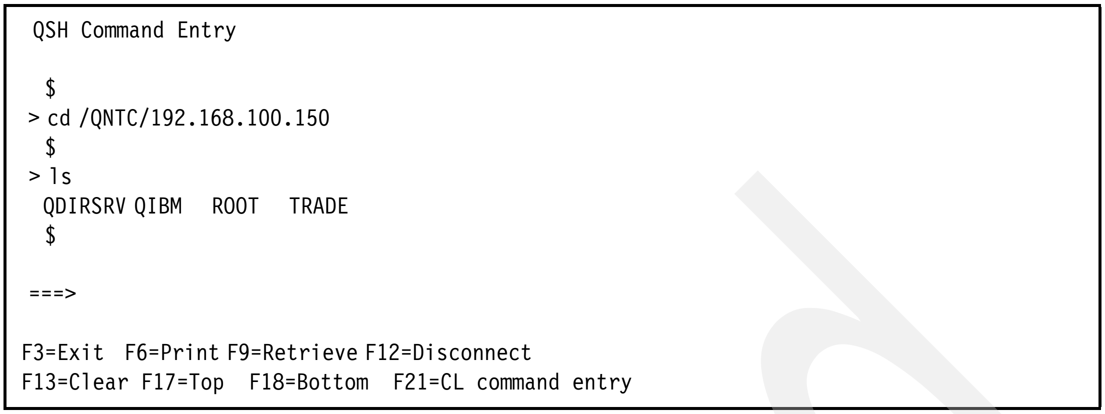
   ```
   cd /QNTC/192.168.100.150
   ```

4. Display the files and directories.

   ```
   ls
   ```

5. You should see output similar to the example shown in Figure 5-85. You should see the names of the file shares defined on the remote system. One of them (TRADE) corresponds to the IASP that we created earlier. Switch to that share:

   ```
   cd trade
   ```



```
 QSH Command Entry

  $
 > cd /QNTC/192.168.100.150
  $
 > ls
  QDIRSRV QIBM   ROOT   TRADE
  $

 ===>

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

Figure 5-85 Remote file system

6. Create a separate directory for the log files.

   ```
   mkdir logs
   ```

7. Make a separate directory for each application server, using the following example:

   ```
   mkdir logs/member1
   mkdir logs/member2
   ```

8. At this point, we have configured the remote access for the transaction log directories from each iSeries where we run WebSphere servers. To access these directories, we use the following path format for each the member, noting the different member number:

   – /QNTC/192.168.100.150/trade/logs/member1
   – /QNTC/192.168.100.150/trade/logs/member2

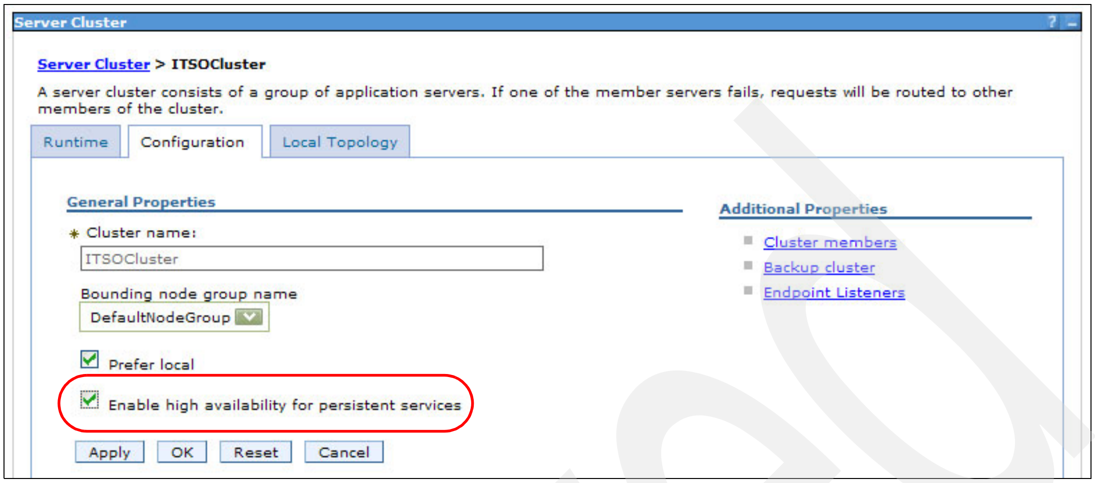
   These are the paths that we'll use to point WebSphere servers to the transaction log files.

Changing WebSphere configuration

When the QNTC file system is configured, we need to update the WebSphere configuration with the new transaction log files location. Perform the following steps for *each* application server in the WebSphere cluster:

1. Access the deployment manager Admin Console.
2. Log in.
3. Expand **Servers** and click **Clusters**.
4. In the next panel, click the link that corresponds to your cluster name.

5. In the cluster's properties panel (Figure 5-86), select the **Enable high availability for persistent services** check box. Click **OK**.



Server Cluster

Server Cluster > ITSOCluster

A server cluster consists of a group of application servers. If one of the member servers fails, requests will be routed to other members of the cluster.

Runtime | Configuration | Local Topology

General Properties

* Cluster name:

ITSOCluster

Bounding node group name

DefaultNodeGroup

☑ Prefer local

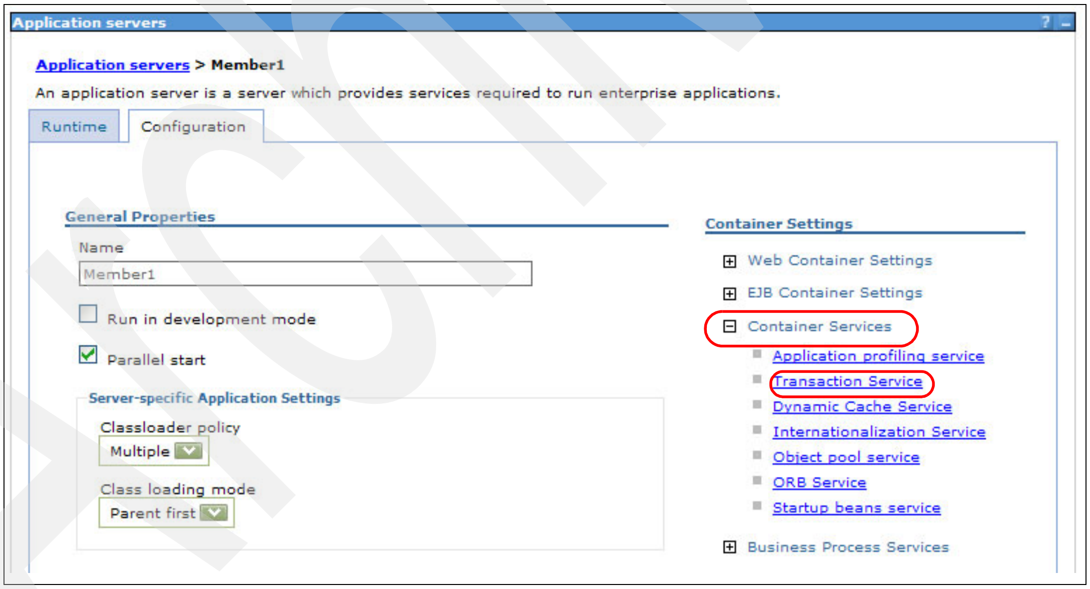
☑ Enable high availability for persistent services

Additional Properties

- Cluster members
- Backup cluster
- Endpoint Listeners

Figure 5-86 Enabling the transaction log failover support

6. In the navigation pane, under Servers, click **Application servers**.
7. Click the first application server in the cluster (**Member1** in our example).
8. In the next panel (Figure 5-87), under Container Settings, expand **Container Services**, and click **Transaction Service**.



Application servers

Application servers > Member1

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

Runtime | Configuration

General Properties

Name

Member1

☐ Run in development mode

☑ Parallel start

Server-specific Application Settings

Classloader policy

Multiple

Class loading mode

Parent first

Container Settings

⊞ Web Container Settings

⊞ EJB Container Settings

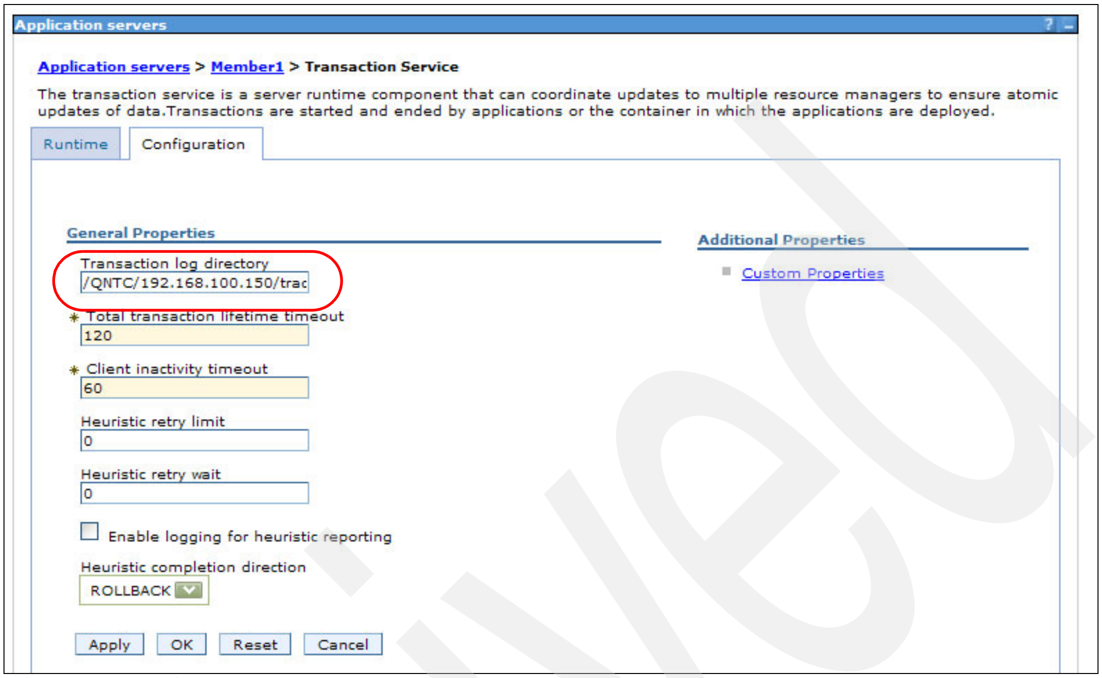
⊟ Container Services

- Application profiling service
- Transaction Service
- Dynamic Cache Service
- Internationalization Service
- Object pool service
- ORB Service
- Startup beans service

⊞ Business Process Services

Figure 5-87 Accessing the transaction services properties

9. In the next panel (Figure 5-88), set the Transaction log directory property. In our example, we typed `/QNTC/192.168.100.150/trade/logs/member1`. Click **OK**.



Application servers > Member1 > **Transaction Service**

The transaction service is a server runtime component that can coordinate updates to multiple resource managers to ensure atomic updates of data.Transactions are started and ended by applications or the container in which the applications are deployed.

General Properties

Transaction log directory: /QNTC/192.168.100.150/trac

* Total transaction lifetime timeout: 120

* Client inactivity timeout: 60

Heuristic retry limit: 0

Heuristic retry wait: 0

□ Enable logging for heuristic reporting

Heuristic completion direction: ROLLBACK

Figure 5-88 Setting the transaction log directory

10. Repeat steps 8 and 9 for *each* cluster member. Make sure you modify the transaction log path in step 9 according to that cluster member name. In our example, for Member2 the path is `/QNTC/192.168.100.150/trade/logs/member2`.

11. Save your changes.

Miscellaneous, but important, steps

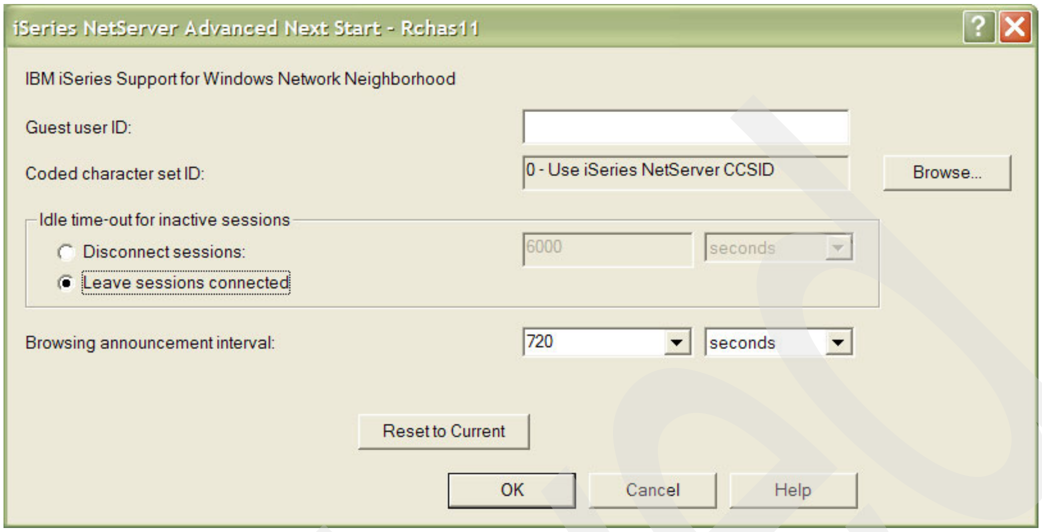
The last set of instructions to finish the transaction log configuration is to set up a special user ID, QEJBSVR, to access the transaction log files. This user profile is created on the system when you install WebSphere Application Server. By moving the transaction log to a remote location, you're required to perform the following steps:

1. Assign a password to the QEJBSVR user profile on *all* systems where you run WebSphere cluster nodes (by default, QEJBSVR doesn't have a password).
2. Create the QEJBSVR user profile on both systems that are part of XSM. QEJBSVR has to have the same password as on other systems (see previous step).
3. On XSM systems, explicitly give a permission for the QEJBSVR user profile to access the transaction log subdirectories. In our example, they are /trade/logs/member1, /trade/logs/member2, and so on.
4. With regard to the remote iSeries file server systems hosting the transaction logs, change the *time-out period* property for iSeries NetServer™ on the two i5/OS cluster node systems to *not* allow idle connections to be terminated by NetServer. You can make this change by using iSeries Navigator.

   a. Open the iSeries NetServer Properties window. Click **iSeries NetServer** → **Properties**.

   b. Click the **Advanced** tab. Click the **Next Start** button.

c. In the iSeries NetServer Advanced Next Start window (Figure 5-89), select the **Leave sessions connected** radio button. Click **OK**.



iSeries NetServer Advanced Next Start - Rchas11

IBM iSeries Support for Windows Network Neighborhood

Guest user ID:

Coded character set ID: 0 - Use iSeries NetServer CCSID

Idle time-out for inactive sessions

○ Disconnect sessions: 6000 seconds

● Leave sessions connected

Browsing announcement interval: 720 seconds

Figure 5-89 Changing NetServer properties

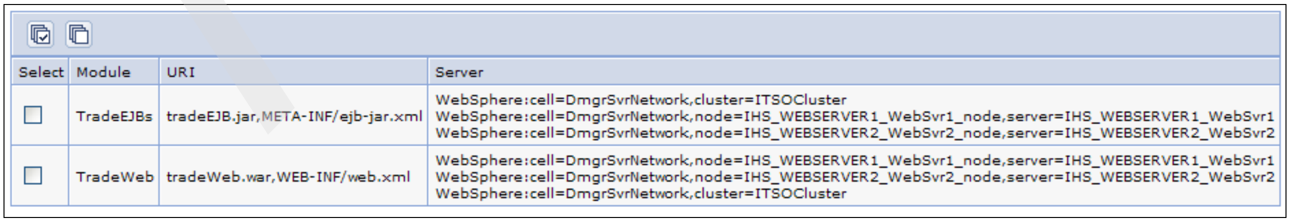
d. Click **OK** again.

e. Restart iSeries NetServer to make the change take effect.

Generating the plug-in configuration file

After a Web server definition and virtual hosts are configured, generate the plug-in configuration file. The plug-in configuration file is needed by each Web server instance. In this scenario, a Web server is installed on each of two different iSeries servers.

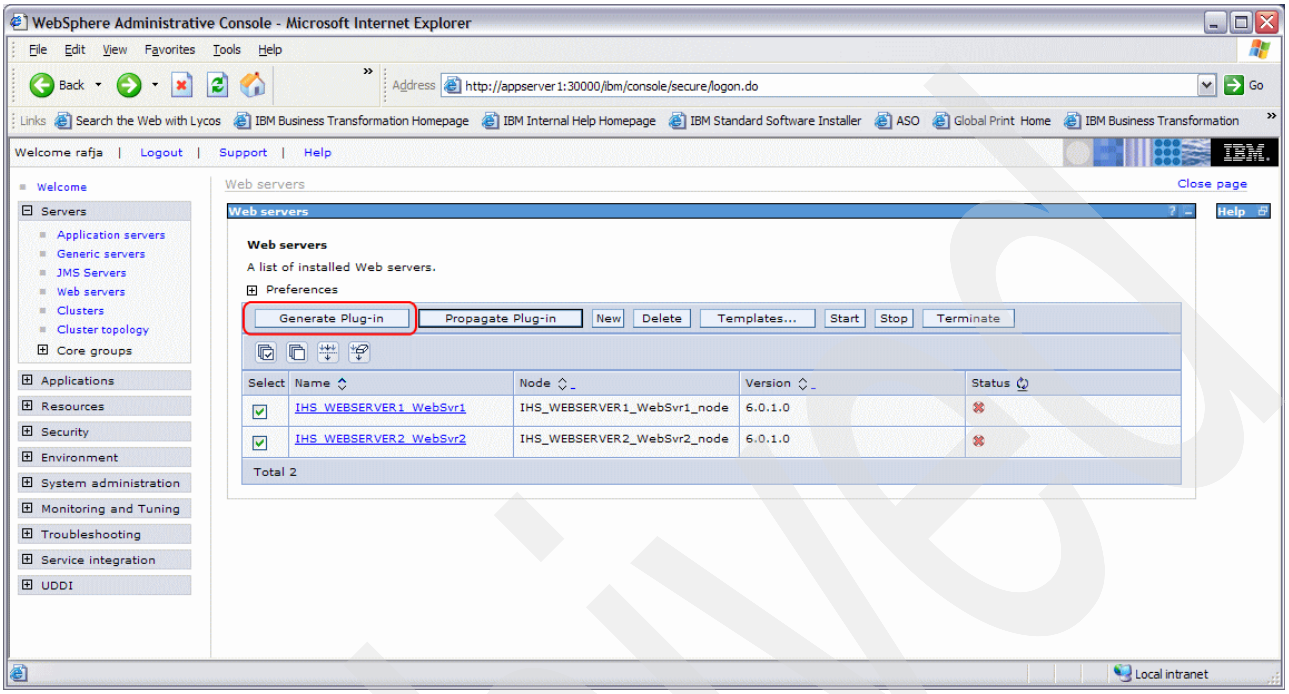
> **Note:** Before you generate the plug-in configuration file, verify that your application maps to the Web server.
>
> 1. In the administrative console, expand **Applications** and click **Enterprise Applications**.
> 2. In the next window, click the link for your application (**Trade** for our example).
> 3. On the application Properties' page, in the Additional Properties section, click **Map modules to servers**.
> 4. The next window indicates whether the Web module or modules and Enterprise JavaBean (EJB) JAR files have been correctly mapped (see Figure 5-90). Map these items to the Web servers if needed.



Select	Module	URI	Server
□	TradeEJBs	tradeEJB.jar,META-INF/ejb-jar.xml	WebSphere:cell=DmgrSvrNetwork,cluster=ITSOCluster WebSphere:cell=DmgrSvrNetwork,node=IHS_WEBSERVER1_WebSvr1_node,server=IHS_WEBSERVER1_WebSvr1 WebSphere:cell=DmgrSvrNetwork,node=IHS_WEBSERVER2_WebSvr2_node,server=IHS_WEBSERVER2_WebSvr2
□	TradeWeb	tradeWeb.war,WEB-INF/web.xml	WebSphere:cell=DmgrSvrNetwork,node=IHS_WEBSERVER1_WebSvr1_node,server=IHS_WEBSERVER1_WebSvr1 WebSphere:cell=DmgrSvrNetwork,node=IHS_WEBSERVER2_WebSvr2_node,server=IHS_WEBSERVER2_WebSvr2 WebSphere:cell=DmgrSvrNetwork,cluster=ITSOCluster

Figure 5-90 Mapping application modules

Continue in the deployment manager WebSphere Administrative Console as shown in Figure 5-77 on page 157. To generate the plug-in configuration file, select both Web server definitions and click **Generate Plug-in**, as shown in Figure 5-91.


WebSphere Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://appserver1:30000/ibm/console/secure/logon.do
Go
Links Search the Web with Lycos IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer ASO Global Print Home IBM Business Transformation
Welcome rafja | Logout | Support | Help
Welcome
Servers
Application servers
Generic servers
JMS Servers
Web servers
Clusters
Cluster topology
Core groups
Applications
Resources
Security
Environment
System administration
Monitoring and Tuning
Troubleshooting
Service integration
UDDI
Web servers
Close page
Help
Preferences
Generate Plug-in
Propagate Plug-in
New
Delete
Templates...
Start
Stop
Terminate
Local intranet

Web servers

A list of installed Web servers.

Select	Name	Node	Version	Status
☑	IHS_WEBSERVER1_WebSvr1	IHS_WEBSERVER1_WebSvr1_node	6.0.1.0	✖
☑	IHS_WEBSERVER2_WebSvr2	IHS_WEBSERVER2_WebSvr2_node	6.0.1.0	✖

Total 2

Figure 5-91 Generate HTTP server plug-in

This action generates two plug-in configuration files, one for each Web server. The WebSphere Administrative Console displays the location of each file. The location is determined by the deployment manager profile name and the Web server definition name.

- The first plug-in configuration file corresponds to the *WebSvr1* instance. The file is located on the deployment manager iSeries (APPSERVER1) in /QIBM/UserData/WebSphere/AppServer/V6/ND/profiles/DmgrSvr/config/cells/DmgrSvrNetwork/nodes/IHS_WEBSERVER1_WebSvr1_node/servers/IHS_WEBSERVER1_WebSvr1/plugin-cfg.xml.
- The second plug-in configuration file corresponds to the *WebSvr2* instance. The file is also located on the deployment manager iSeries (APPSERVER1) in /QIBM/UserData/WebSphere /AppServer/V6/ND/profiles/DmgrSvr/config/cells/DmgrSvrNetwork/nodes /IHS_WEBSERVER2_WebSvr2_node/servers/IHS_WEBSERVER2_WebSvr2/plugin-cfg.xml.

Now you need to move both plug-in configuration files to the Web server iSeries systems.

Moving the plug-in configuration file

The deployment manager iSeries system generates the plug-in configuration files. The files are needed on the Web server iSeries specifically for each Web server instance. In this scenario with the Web servers and application servers residing on different iSeries, move the plug-in configuration files from the deployment manager system (APPSERVER1) to each of the Web server iSeries systems.

To successfully move the configuration file, you need to know the file's current location and the destination directory. The current location is specified during the plug-in generation step. Refer "Generating the plug-in configuration file" on page 166 for the current locations.

The destination directories are located on the Web server iSeries systems. For both the *WebSvr1* instance and the *WebSvr2* instance, the destination directory is the same for each Web server iSeries. The directory is /QIBM/UserData/WebSphere/AppServer/V6/ND/profiles/*profile-name*/config, where *profile-name* is DmgrSvr.

After you know the source and destination directories, you can move each specific plug-in configuration file to the appropriate Web server iSeries using any method such as File Transfer Protocol (FTP), mapped drive, and so on.

Attention: After you move the plug-in configuration file to the Web server iSeries, check the authorities of this file. Run the following OS/400 command on both the WEBSERVER1 and WEBSERVER2 iSeries, to check the current authorities of the plugin-cfg.xml file:

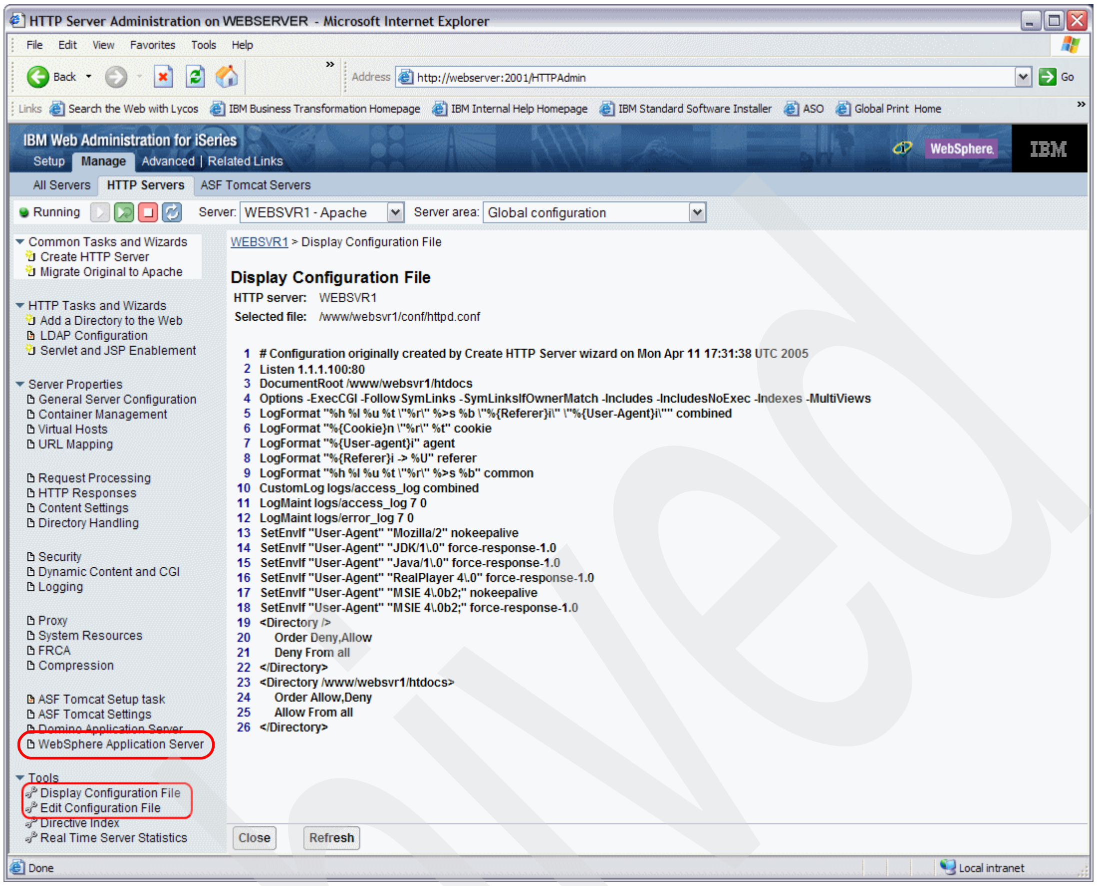
```
WRKAUT OBJ('QIBM/UserData/WebSphere/AppServer/V6/ND/profiles/profileName/config/
plugin-cfg.xml')
```

Ensure the QTMHHTTP user has read (R) authority. If the user does not have this authority, add it manually.

Configuring HTTP Server plug-in

To establish communication between the Web server and the application server, the Web server needs to know where to find the plug-in service program and the plug-in configuration file. To configure the plug-in component, use Web Administration for iSeries.

1. Access the IBM Web Administration for iSeries GUI. Refer to “Starting IBM Web Administration for iSeries” on page 68.
2. After you log into the GUI, click the **Manage** tab and click the specific HTTP server used in this configuration.
3. You see a window like the example in Figure 5-92. Click **WebSphere Application Server** at the bottom of the navigation pane.


HTTP Server Administration on WEBSERVER - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://webserver:2001/HTTPAdmin
Go
Links Search the Web with Lycos IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer ASO Global Print Home
IBM Web Administration for iSeries
Setup Manage Advanced | Related Links
WebSphere
IBM
All Servers HTTP Servers ASF Tomcat Servers
Running
Server: WEBSVR1 - Apache
Server area: Global configuration
Common Tasks and Wizards
Create HTTP Server
Migrate Original to Apache
HTTP Tasks and Wizards
Add a Directory to the Web
LDAP Configuration
Servlet and JSP Enablement
Server Properties
General Server Configuration
Container Management
Virtual Hosts
URL Mapping
Request Processing
HTTP Responses
Content Settings
Directory Handling
Security
Dynamic Content and CGI
Logging
Proxy
System Resources
FRCA
Compression
ASF Tomcat Setup task
ASF Tomcat Settings
Domino Application Server
WebSphere Application Server
Tools
Display Configuration File
Edit Configuration File
Directive Index
Real Time Server Statistics
WEBSVR1 > Display Configuration File
Close
Refresh
Done
Local intranet

Display Configuration File

HTTP server: WEBSVR1

Selected file: /www/websvr1/conf/httpd.conf

```
# Configuration originally created by Create HTTP Server wizard on Mon Apr 11 17:31:38 UTC 2005
Listen 1.1.1.100:80
DocumentRoot /www/websvr1/htdocs
Options -ExecCGI -FollowSymLinks -SymLinksIfOwnerMatch -Includes -IncludesNoExec -Indexes -MultiViews
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%{Cookie}n \"%r\" %t" cookie
LogFormat "%{User-agent}i" agent
LogFormat "%{Referer}i -> %U" referer
LogFormat "%h %l %u %t \"%r\" %>s %b" common
CustomLog logs/access_log combined
LogMaint logs/access_log 7 0
LogMaint logs/error_log 7 0
SetEnvIf "User-Agent" "Mozilla/2" nokeepalive
SetEnvIf "User-Agent" "JDK/1\.0" force-response-1.0
SetEnvIf "User-Agent" "Java/1\.0" force-response-1.0
SetEnvIf "User-Agent" "RealPlayer 4\.0" force-response-1.0
SetEnvIf "User-Agent" "MSIE 4\.0b2;" nokeepalive
SetEnvIf "User-Agent" "MSIE 4\.0b2;" force-response-1.0
<Directory />
   Order Deny,Allow
   Deny From all
</Directory>
<Directory /www/websvr1/htdocs>
   Order Allow,Deny
   Allow From all
</Directory>
```

Figure 5-92 HTTP Server instance configuration file

4. The WebSphere Application Server configuration pane opens.

   a. Click the radio button next to your version and edition of WebSphere Application Server (in this example, it is the ND edition).

   b. A drop-down field is displayed. Select your WebSphere Application Server profile name from the list. In our case, the name is **DmgrSvr**.

   c. Click **OK**.

5. In the navigation pane, click the **Display Configuration File** link.

6. The first two lines in the configuration file should look similar to this:

   ```
   WebSpherePluginConfig /QIBM/UserData/WebSphere/AppServer/V6/ND/profiles
   /profileName/config/plugin-cfg.xml

   LoadModule was_ap20_module /QSYS.LIB/QWAS6.LIB/QSVTAP20.SRVPGM
   ```

   Here *profileName* is the name of the HTTP Server profile that you created in 5.5.2, “Creating HTTP Server instances” on page 140.

   The first line instructs the HTTP Server instance where to find the plug-in configuration file. The second line is an HTTP server command which loads the Web server plug-in to the HTTP Web server instance. Click **OK** to save the changes.

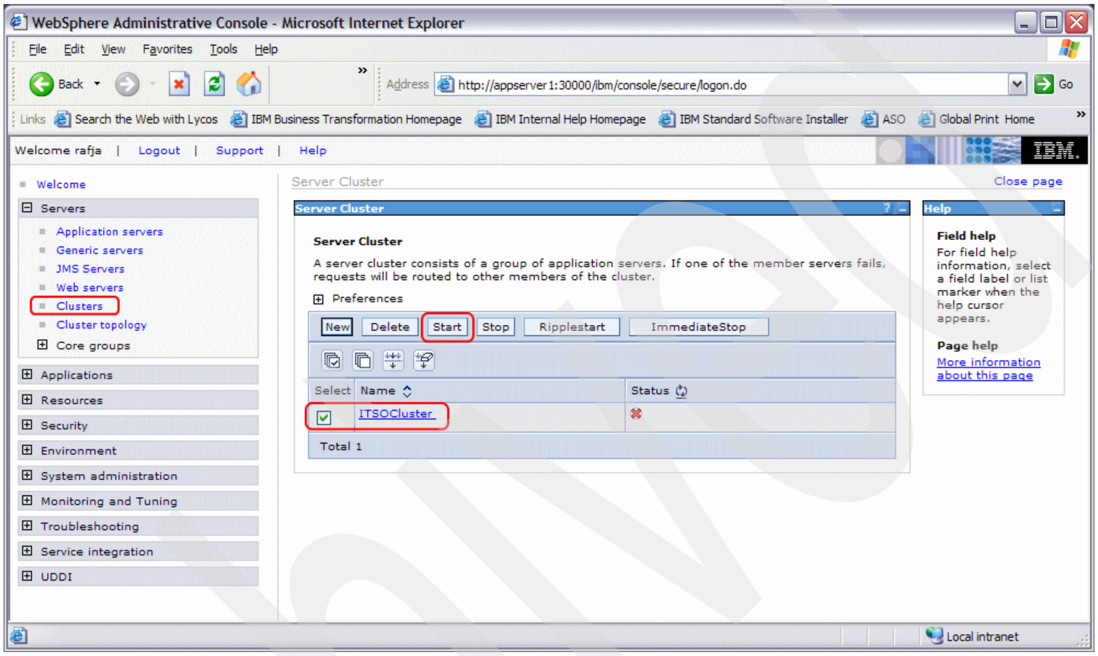
7. Repeat steps 1 through 6 on the second HTTP server system.

At this point, both Web servers are fully configured and ready to be used in the topology.

Starting the cluster and HTTP Server instances

To start the WebSphere Application Server cluster, you need to start WebSphere Administration Console for deployment manager as explained in step 6 on page 143.

1. Log into the console.
2. In the left navigation area, click **Servers** → **Clusters**.
3. In the Server Cluster pane show on the right in Figure 5-93, select **ITSOCluster** and click **Start**.


WebSphere Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://appserver1:30000/ibm/console/secure/logon.do
Go
Links Search the Web with Lycos IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer ASO Global Print Home
Welcome rafja | Logout | Support | Help
IBM.
Welcome
Servers
Application servers
Generic servers
JMS Servers
Web servers
Clusters
Cluster topology
Core groups
Applications
Resources
Security
Environment
System administration
Monitoring and Tuning
Troubleshooting
Service integration
UDDI
Server Cluster
Close page
Server Cluster
Preferences
New Delete Start Stop Ripplestart ImmediateStop
Select Name Status
ITSOCluster
Total 1
Help
Field help
Page help
More information about this page
Local intranet

Server Cluster

A server cluster consists of a group of application servers. If one of the member servers fails, requests will be routed to other members of the cluster.

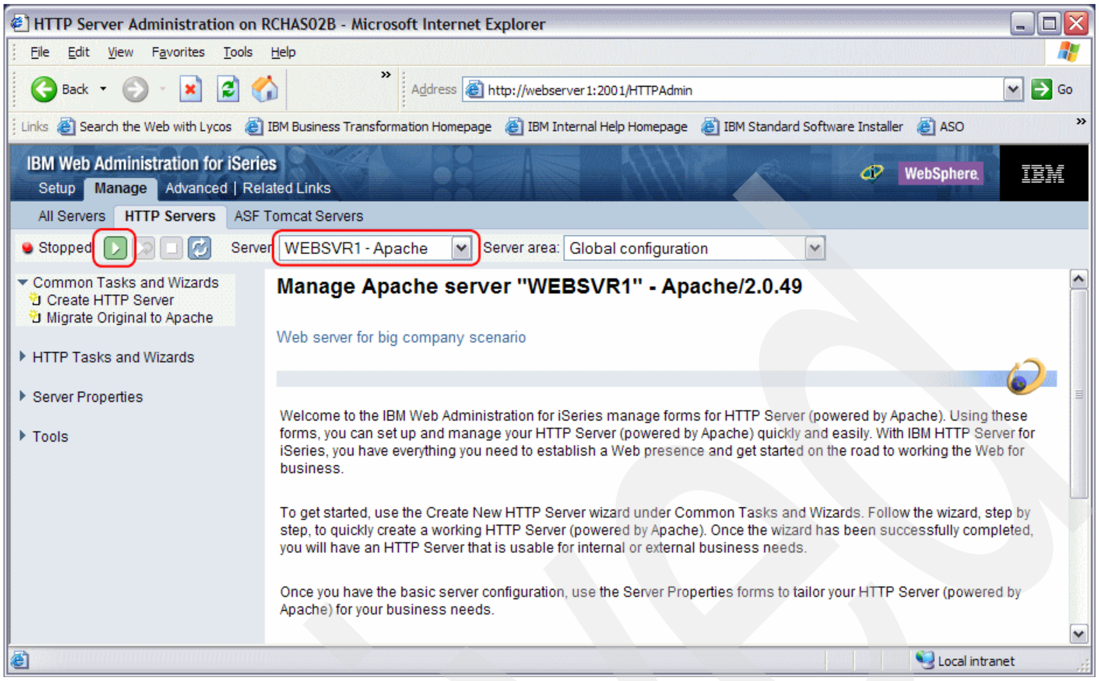
For field help information, select a field label or list marker when the help cursor appears.

Figure 5-93 Starting the WebSphere cluster

When the cluster start is completed, the status icon changes to a green arrow. When the cluster has started, the Web server instances can be started.

Start the Web server instances from the IBM Web Administration for iSeries GUI.

1. Open the GUI for both the WEBSERVER1 and WEBSERVER2 iSeries.
2. Go to the **Manage** tab and select the appropriate Web instance (WebSvr1 or WebSvr2).
3. To start the Web server instance, click the green button as shown in Figure 5-94.


HTTP Server Administration on RCHAS02B - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://webserver1:2001/HTTPAdmin
Go
Links Search the Web with Lycos IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer ASO
IBM Web Administration for iSeries
Setup Manage Advanced | Related Links
WebSphere
IBM
All Servers HTTP Servers ASF Tomcat Servers
Stopped
Server: WEBSVR1 - Apache
Server area: Global configuration
Common Tasks and Wizards
Create HTTP Server
Migrate Original to Apache
HTTP Tasks and Wizards
Server Properties
Tools
Local intranet

Manage Apache server "WEBSVR1" - Apache/2.0.49

Web server for big company scenario

Welcome to the IBM Web Administration for iSeries manage forms for HTTP Server (powered by Apache). Using these forms, you can set up and manage your HTTP Server (powered by Apache) quickly and easily. With IBM HTTP Server for iSeries, you have everything you need to establish a Web presence and get started on the road to working the Web for business.

To get started, use the Create New HTTP Server wizard under Common Tasks and Wizards. Follow the wizard, step by step, to quickly create a working HTTP Server (powered by Apache). Once the wizard has been successfully completed, you will have an HTTP Server that is usable for internal or external business needs.

Once you have the basic server configuration, use the Server Properties forms to tailor your HTTP Server (powered by Apache) for your business needs.

Figure 5-94 Starting the HTTP server instance

After your environment is started, you can test it.

5.6 Configuring Load Balancer

Now that we have successfully configured WebSphere Application Server, Web servers, and database, it's time to add more components to our configuration. Let's start with Load Balancer.

In our scenario, we try to eliminate the Network Dispatcher component as a single point of failure in the network. To resolve this, we use a second Network Dispatcher server that monitors the primary server and stands by to take over the task of load balancing should the primary server fail at any time. Figure 5-95 shows the high level design for a high availability Network Dispatcher configuration.

Note: Throughout the book, we use two terms interchangeably: Network Dispatcher and Load Balancer. These two terms refer to the same component.

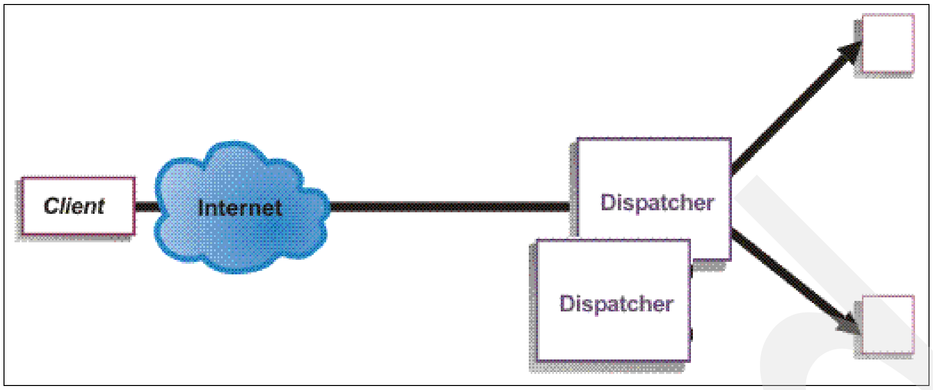


Figure 5-95 High level design for a high availability dispatcher configuration

Installing Load Balancer

Read Appendix D, "Installing Load Balancer" on page 281, for instructions on how to install Load Balancer.

5.6.1 Configuring the Load Balancer: A simple scenario

This scenario, as shown in Figure 5-96, represents Load Balancer configured on one server only and load balancing the traffic between two Web servers.

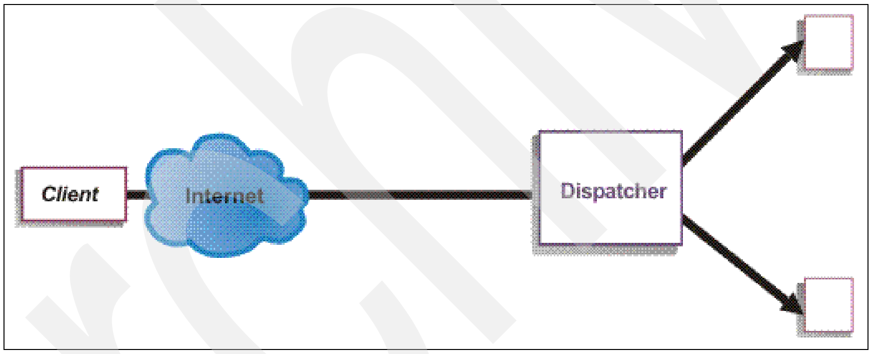


Figure 5-96 Load Balancer simple scenario

To configure Load Balancer on one node, follow these steps:

1. Verify that the Dispatcher server has been started (on Windows 2000 server, the Dispatcher server runs as a service that starts automatically). Type the following command in a command window:

   ```
   dsserver
   ```

2. Start the Load Balancer GUI by selecting **Start** → **Programs** → **IBM WebSphere** → **Edge Components** → **Load Balancer** → **Load Balancer**.

   The Load Balancer GUI is a Java client that can also be installed on a client server, so the administrator can work remotely.

3. When the Load Balancer Administration Tools window (Figure 5-97) opens, right-click **Dispatcher** in the left pane and select **Connect to Host**.

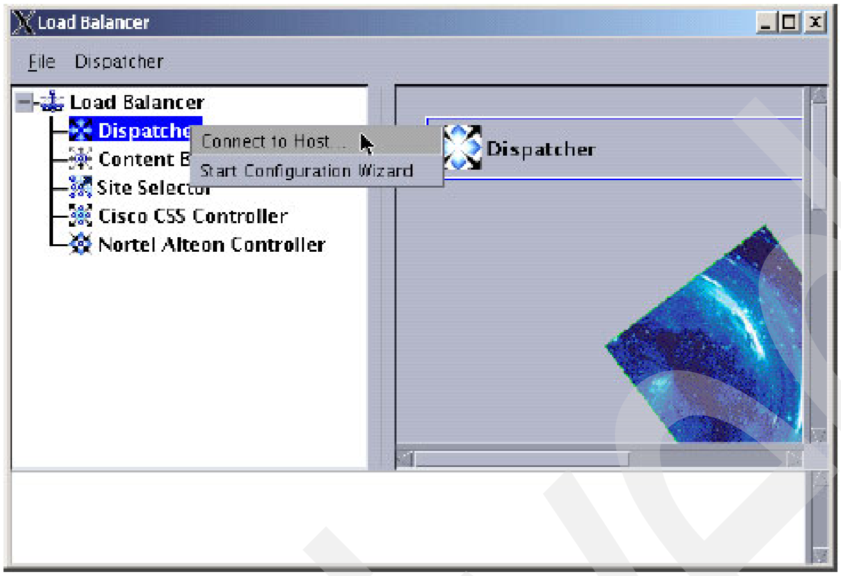
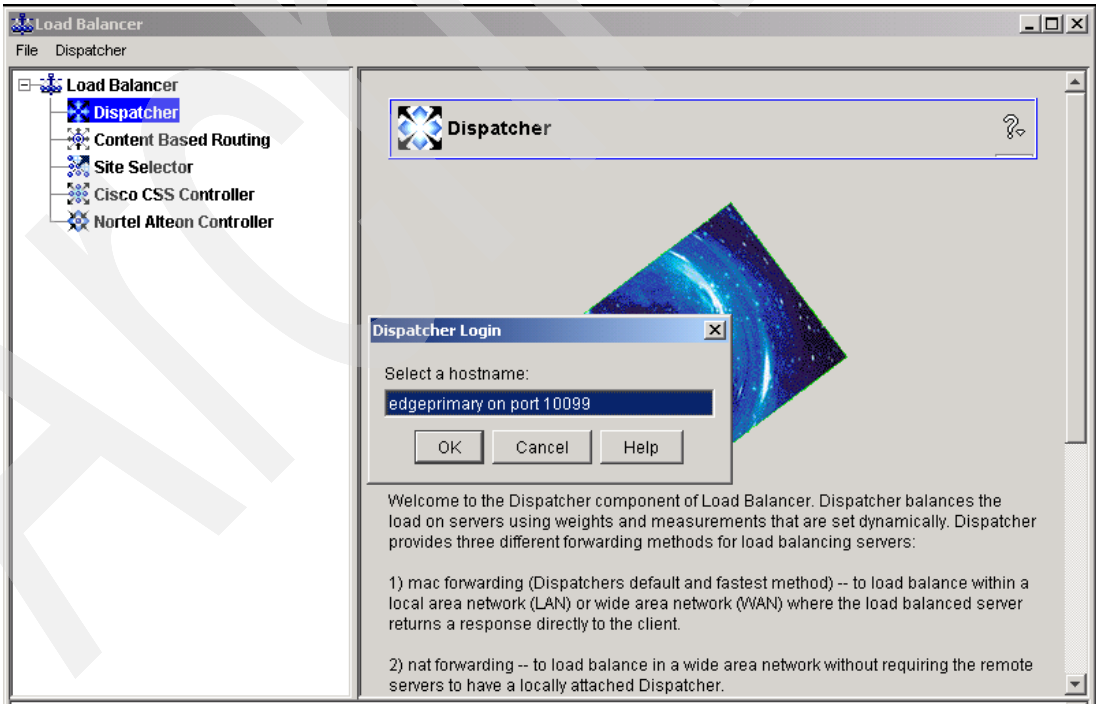


Figure 5-97 Load Balancer Administrator Tools

4. A pop-up window (see Figure 5-98) opens, prompting you for the Load Balancer server to connect to. Be default, it populates the text box with the host name of the local workstation and port number. Accept the defaults and click **OK**.



Select a hostname:
edgeprimary on port 10099

Welcome to the Dispatcher component of Load Balancer. Dispatcher balances the load on servers using weights and measurements that are set dynamically. Dispatcher provides three different forwarding methods for load balancing servers:

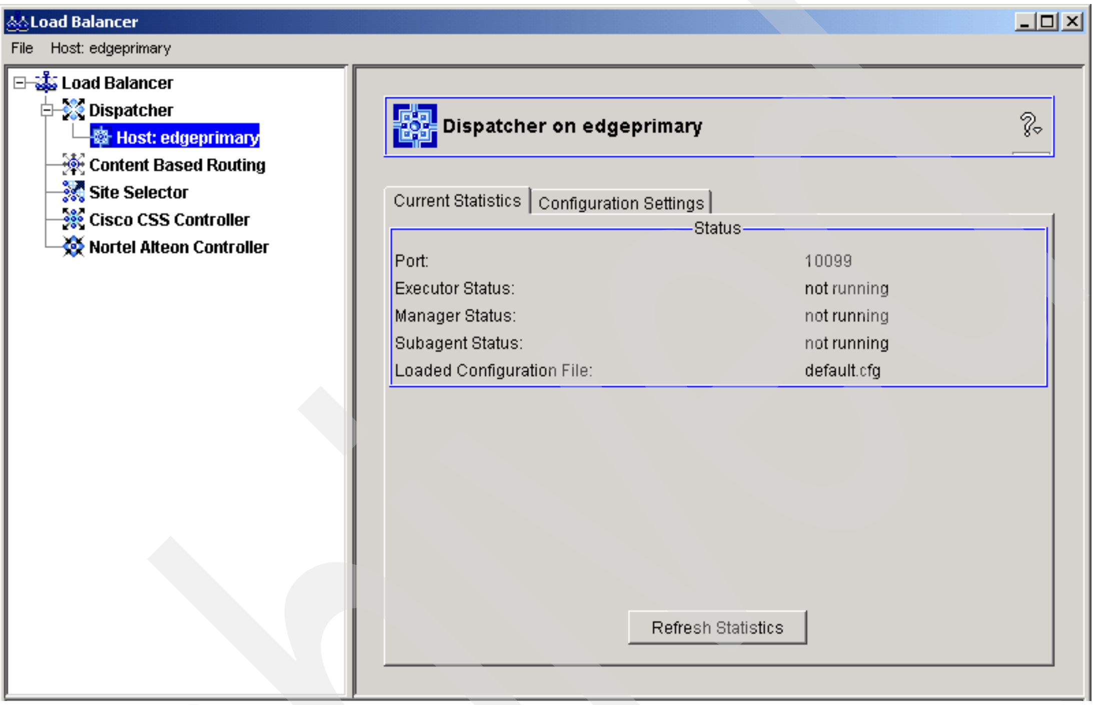
1) mac forwarding (Dispatchers default and fastest method) -- to load balance within a local area network (LAN) or wide area network (WAN) where the load balanced server returns a response directly to the client.

2) nat forwarding -- to load balance in a wide area network without requiring the remote servers to have a locally attached Dispatcher.

Figure 5-98 Dispatcher Login window

5. After connecting to the Load Balancer server, a new entry is added to the GUI window in the left pane (see Figure 5-99). This entry contains the host name of the selected server, which in this example is Host: edgeprimary. From this point forward, the entire configuration that we do is added to this element in a tree structure.

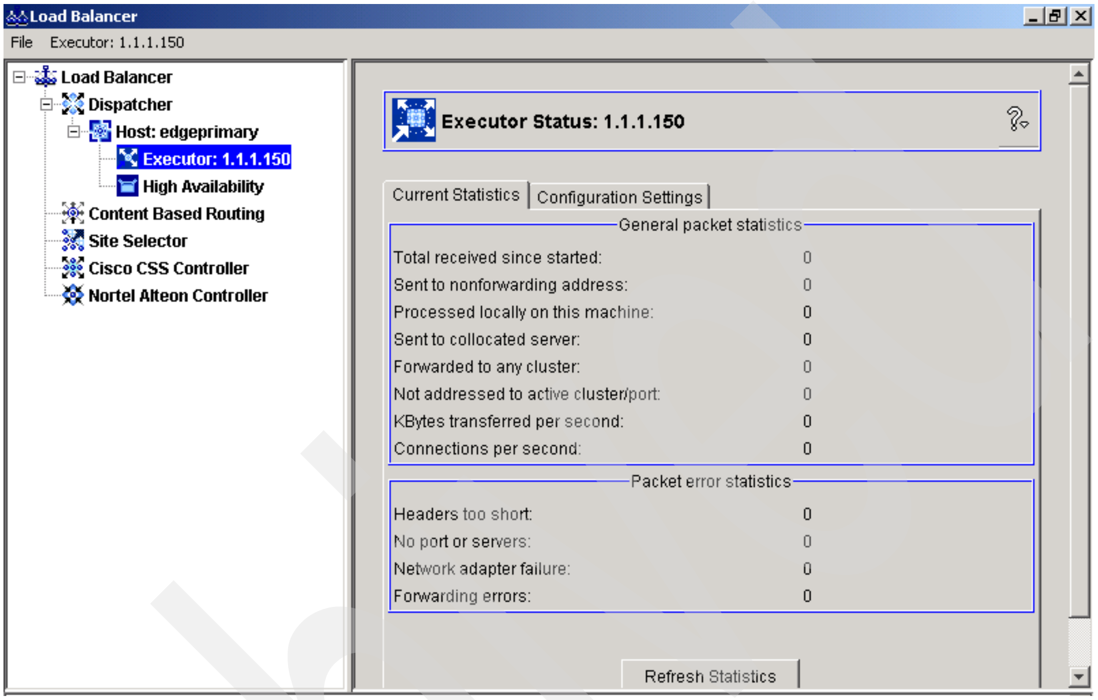
   Start the Executor component, which is the component that distributes the load to the servers. Right-click **Host: edgeprimary** and select **Start Executor**.



Port: 10099
Executor Status: not running
Manager Status: not running
Subagent Status: not running
Loaded Configuration File: default.cfg

Figure 5-99 Tree structure of Administration Tools

Note: If Executor is started successfully, a new item named “Executor” is added to the left pane. In our scenario, the Load Balancer IP address is 1.1.1.150, so this IP address is shown in this new item as well (see Figure 5-100).



Executor Status: 1.1.1.150

General packet statistics

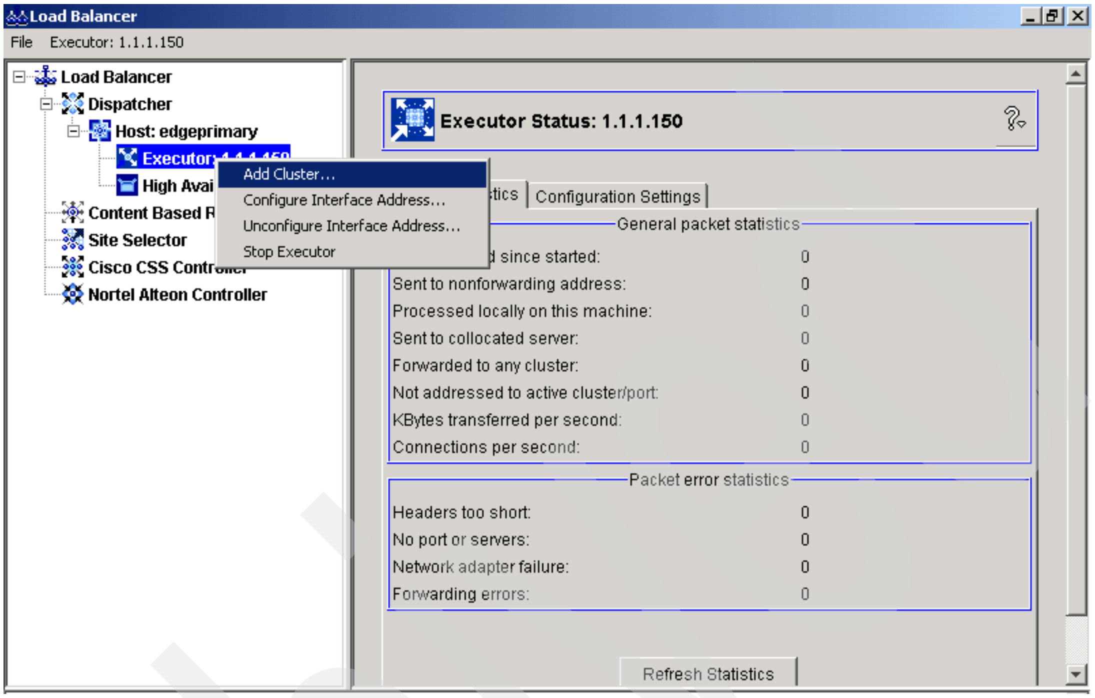
Total received since started:	0
Sent to nonforwarding address:	0
Processed locally on this machine:	0
Sent to collocated server:	0
Forwarded to any cluster:	0
Not addressed to active cluster/port:	0
KBytes transferred per second:	0
Connections per second:	0

Packet error statistics

Headers too short:	0
No port or servers:	0
Network adapter failure:	0
Forwarding errors:	0

Figure 5-100 Starting the Executor component window

6. Add a cluster. In our scenario, we have a cluster called *clusterlb* (1.1.1.100). This cluster contains two Web servers, webserver1 (1.1.1.200) and webserver2 (1.1.1.201). Right-click **Executor: 1.1.1.150** and select **Add Cluster**, as shown in Figure 5-101.



General packet statistics

d since started: 0
Sent to nonforwarding address: 0
Processed locally on this machine: 0
Sent to collocated server: 0
Forwarded to any cluster: 0
Not addressed to active cluster/port: 0
KBytes transferred per second: 0
Connections per second: 0

Packet error statistics

Headers too short: 0
No port or servers: 0
Network adapter failure: 0
Forwarding errors: 0

Figure 5-101 Selecting to add a cluster

7. The Add a cluster window (Figure 5-102) opens, in which you complete the following steps.

> **Note:** We recommend that you use the host name in all fields that require the server identifier.

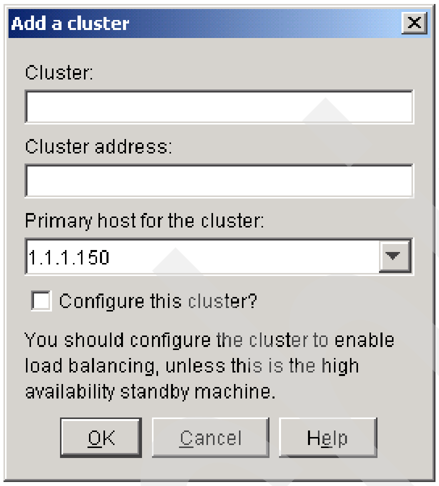
a. In the Cluster field, type the name of the cluster.

b. In the Cluster address field, type the cluster IP address.

c. In the Primary host for the cluster field, select the Load Balancer's IP address.

d. Select the **Configure this cluster?** check box. This option is used to create an IP alias in the operating system for the cluster IP address. You can also deselect this option and add the IP alias manually using operating system tools or commands.

e. Click **OK**.



Add a cluster

Cluster:

Cluster address:

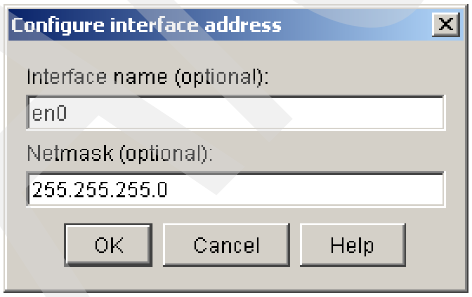
Primary host for the cluster: 1.1.1.150

□ Configure this cluster?

You should configure the cluster to enable load balancing, unless this is the high availability standby machine.

Figure 5-102 Add a cluster window

8. If you selected the *Configure this cluster* check box, the Configure interface address window (Figure 5-103) opens. In the Interface name field, type the interface identification, and in the Netmask field, type the network mask. Click **OK**.



Configure interface address

Interface name (optional): en0

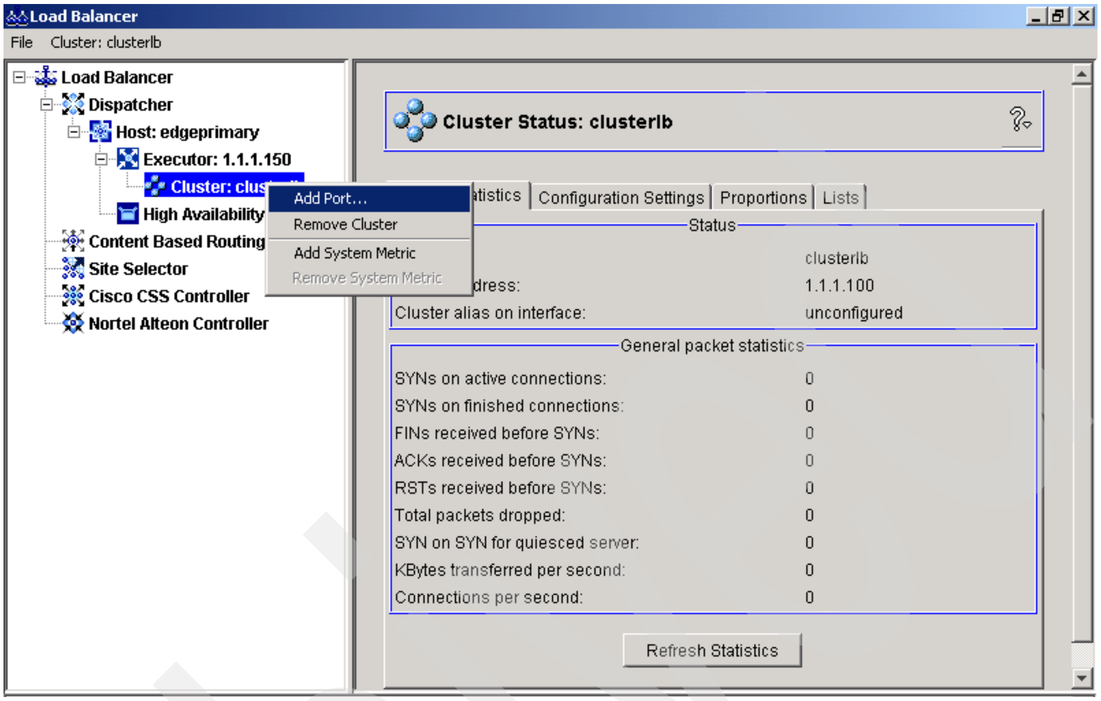
Netmask (optional): 255.255.255.0

Figure 5-103 Configure interface address window

> **Note:** You need to review the interface name because, in some cases, the interface name assigned by system results in errors. In our server, the interface that is associated with the IP address 1.1.1.100 is en0.

9. Add each port that will be load balanced by Dispatcher.

   a. Right-click **Cluster: clusterlb** and select **Add Port** (see Figure 5-104).



clusterlb

dress: 1.1.1.100

Cluster alias on interface: unconfigured

SYNs on active connections: 0

SYNs on finished connections: 0

FINs received before SYNs: 0

ACKs received before SYNs: 0

RSTs received before SYNs: 0

Total packets dropped: 0

SYN on SYN for quiesced server: 0

KBytes transferred per second: 0

Connections per second: 0

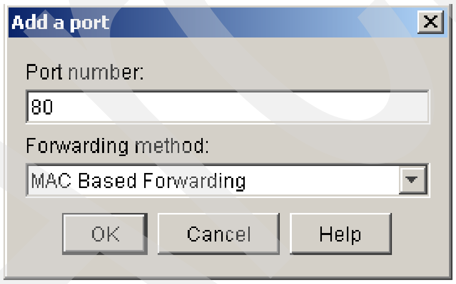
Figure 5-104 Selecting to add a port

   b. The port that we are adding refers to the port that the client will access. In our scenario, we use port 80. In the Add a port window (Figure 5-105), complete these steps:

      i. In the Port number field, type the port number.

      ii. In the Forwarding method field, select **MAC Based Forwarding**.

      iii. Click **OK**.

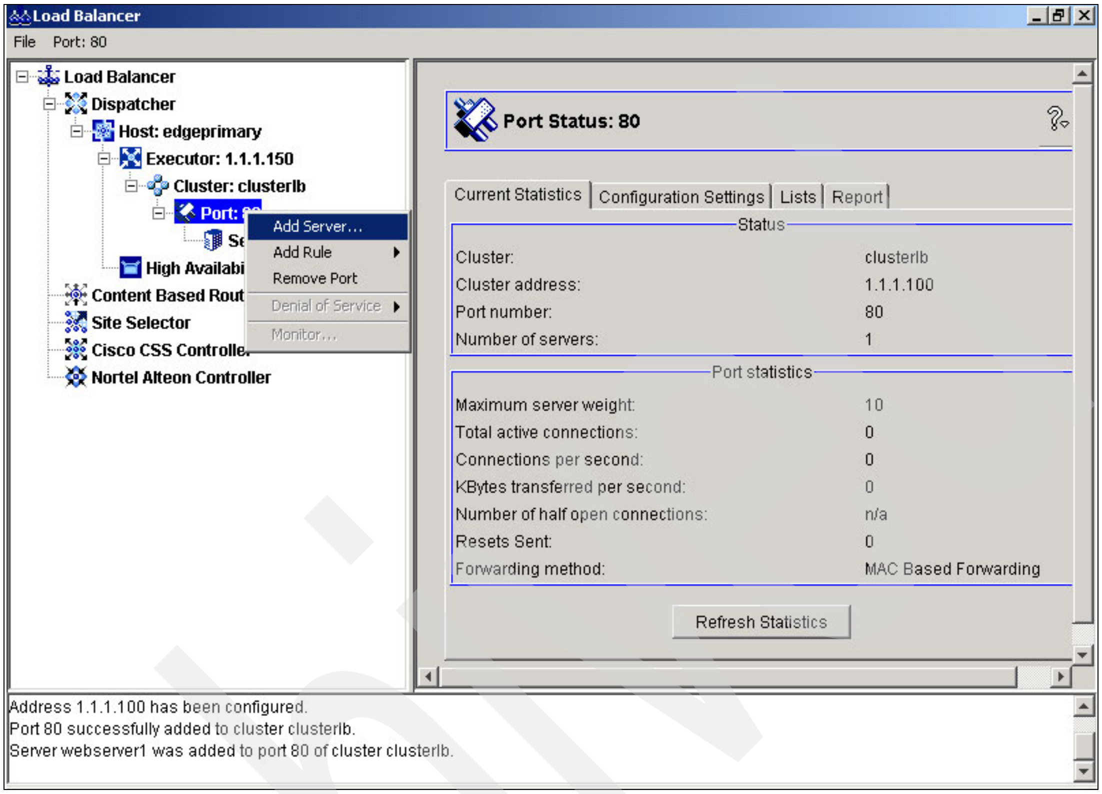


Port number: 80

Forwarding method: MAC Based Forwarding

Figure 5-105 Add a port window

c. A new item representing port 80 is added to the left pane of the GUI. Add the servers that will receive the load for port 80 in cluster clusterlb. Right-click **Port:80** and select **Add Server** as shown in Figure 5-106.



Status

Cluster: clusterlb
Cluster address: 1.1.1.100
Port number: 80
Number of servers: 1

Port statistics

Maximum server weight: 10
Total active connections: 0
Connections per second: 0
KBytes transferred per second: 0
Number of half open connections: n/a
Resets Sent: 0
Forwarding method: MAC Based Forwarding

```
Address 1.1.1.100 has been configured.
Port 80 successfully added to cluster clusterlb.
Server webserver1 was added to port 80 of cluster clusterlb.
```

Figure 5-106 Adding a server

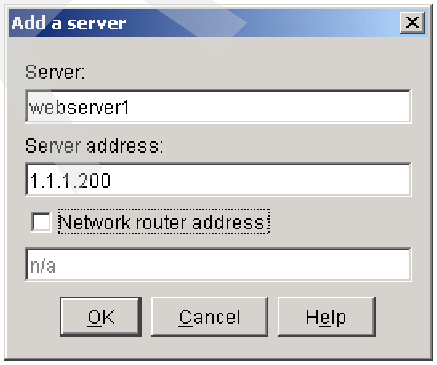
d. The Add a server window (Figure 5-107) opens. It prompts you for information about the first server.

i. In the Server field, type the host name of your Web server.

ii. In the Server address field, type the IP address.

iii. Click **OK**.

> **Note:** The Network router address check box is unavailable because the Load Balancer and HTTP servers are in the same subnet, so the router address is not applicable.



Server: webserver1

Server address: 1.1.1.200

☐ Network router address

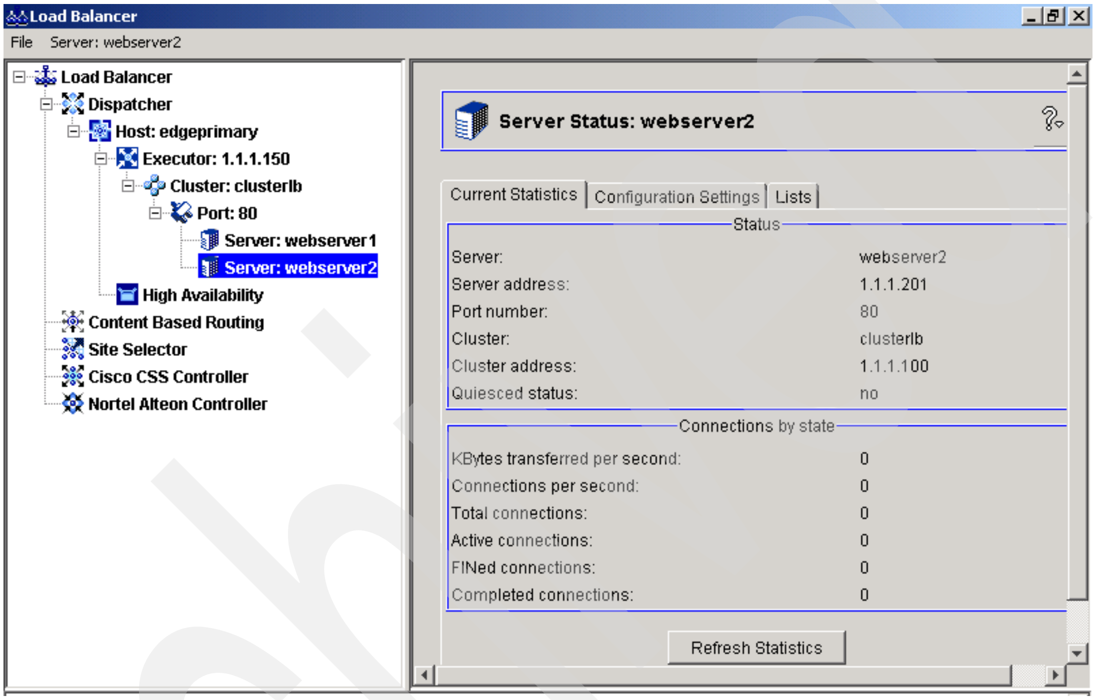
n/a

Figure 5-107 Add a server window

10. Repeat the previous steps to add the second server.

> **Attention:** The first server that we added in our scenario is webserver1, and its IP address is 1.1.1.200. The second server that we add in our scenario is webserver2, and its IP address is 1.1.1.201.

11. The load balancing part of the configuration is done (see Figure 5-108). All the information that the Dispatcher needs to load balance for our cluster is now configured.



Server Status: webserver2

Status

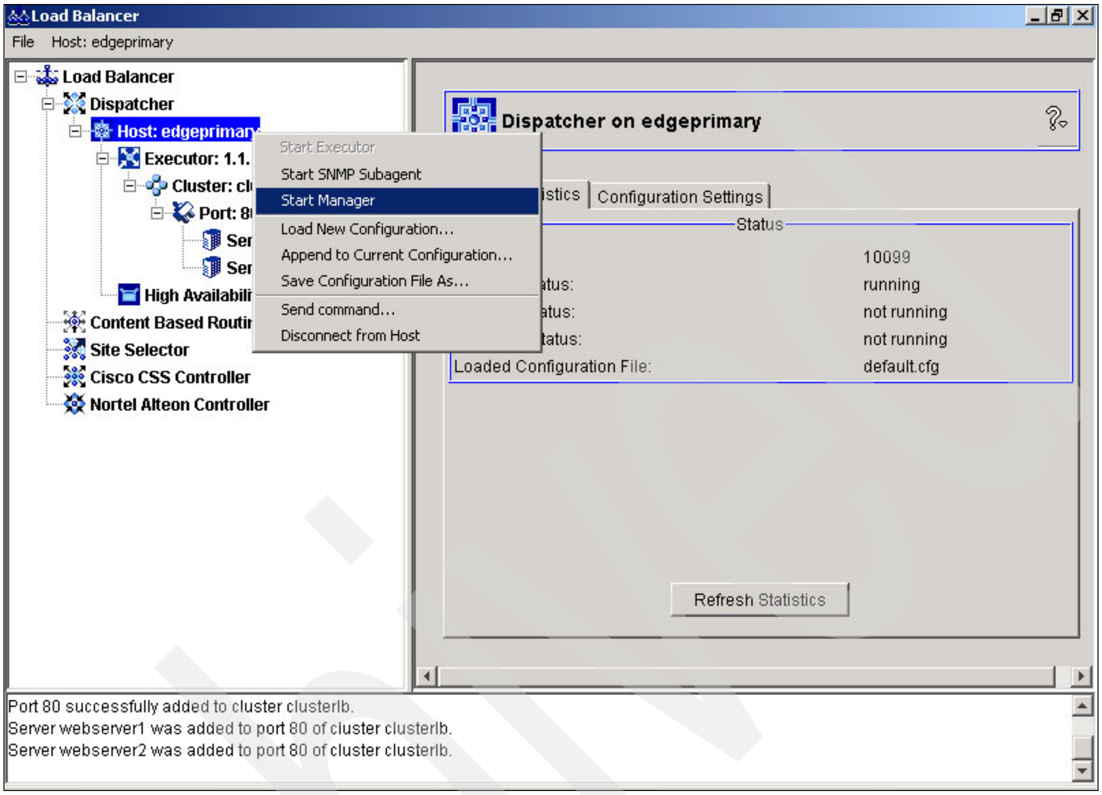
Server: webserver2
Server address: 1.1.1.201
Port number: 80
Cluster: clusterlb
Cluster address: 1.1.1.100
Quiesced status: no

Connections by state

KBytes transferred per second: 0
Connections per second: 0
Total connections: 0
Active connections: 0
FINed connections: 0
Completed connections: 0

Figure 5-108 Load balancer configuration

12. You now need to configure the Manager component to work with dynamic weight values and failure detection. Start the Manager component. Right-click **Host: edgeprimary** and select **Start Manager**, as shown in Figure 5-109.

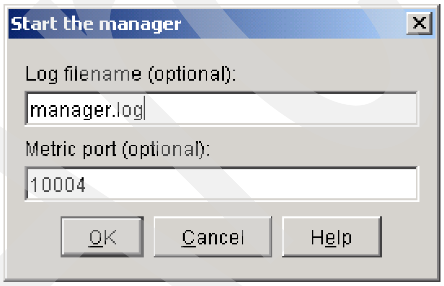


atus: running
atus: not running
tatus: not running
10099
Loaded Configuration File: default.cfg

Port 80 successfully added to cluster clusterlb.
Server webserver1 was added to port 80 of cluster clusterlb.
Server webserver2 was added to port 80 of cluster clusterlb.

Figure 5-109 Starting the Manager

13. The Start the manager window opens (Figure 5-110), in which you can select the name of the Manager log file and the metric port. We leave the default values. Click **OK**.



Log filename (optional): manager.log

Metric port (optional): 10004

Figure 5-110 Starting the manager

14. The Manager needs advisors to generate a weight value based on the response time from each server in the cluster. The advisor is also needed to detect a failure in the service of any balanced server (in our case, a failure in the Web server service).

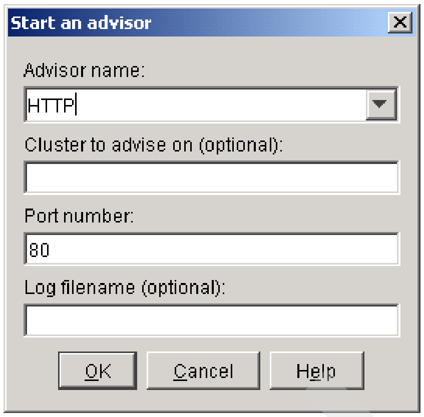
    Due to the importance of the advisor, when you start Manager, the Load Balancer GUI automatically displays a pop-up window prompting you to start an advisor.

    In our scenario, we are load balancing a Web server using the HTTP protocol. Therefore, we use the default values as shown in Figure 5-111. In the Advisor name field, you see `HTTP`, and in the Port number field, you see port `80`. These default values are presented to us because we previously added port 80 in our configuration.

You can also choose a specific cluster to associate this advisor with. By leaving the optional Cluster to advice on field blank, this advisor is automatically associated with all clusters that are load balancing on port 80.

If you want to specify a log file name for this advisor, type in the desired name in the Log filename field. The default file name for the HTTP advisor is Http_80.log.

Click **OK**.



Advisor name: HTTP

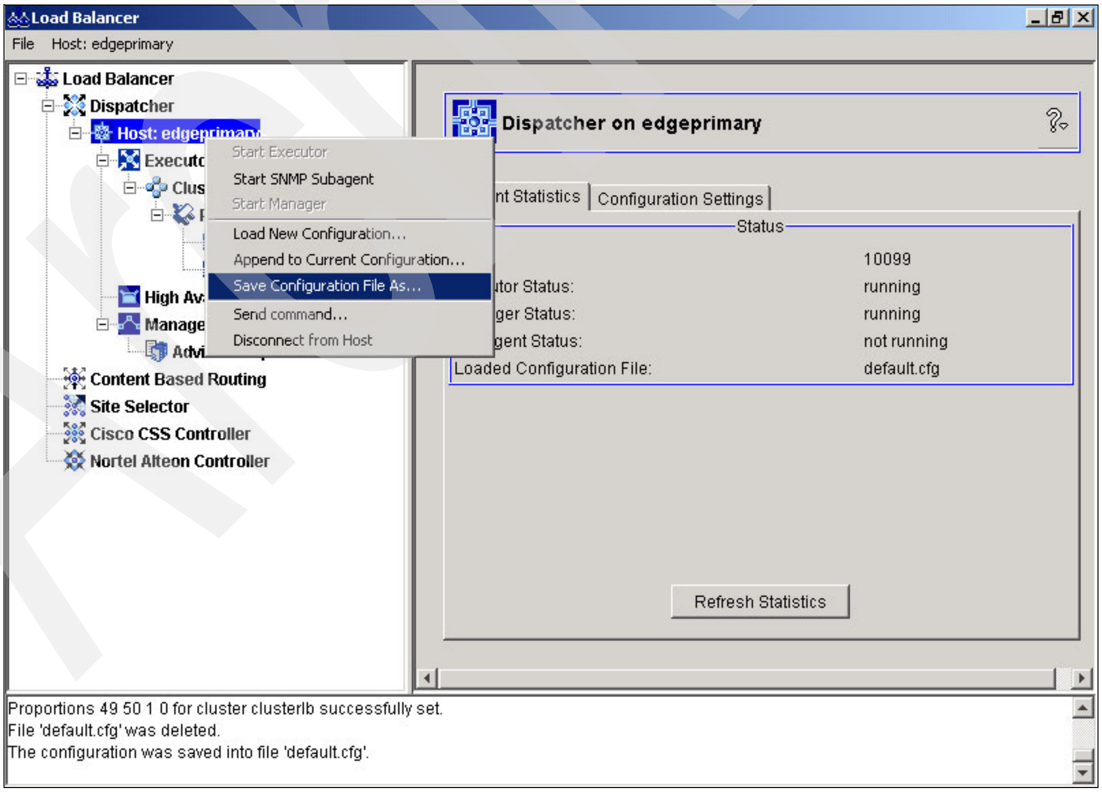
Cluster to advise on (optional):

Port number: 80

Log filename (optional):

Figure 5-111 Start an advisor window

15. We have completed the basic load balancing configuration, so we need to save the configuration file as it is so far. Right-click **Host: clusterlb** and select **Save Configuration File As**, as shown in Figure 5-112.



	10099
tor Status:	running
ger Status:	running
gent Status:	not running
Loaded Configuration File:	default.cfg

```
Proportions 49 50 1 0 for cluster clusterlb successfully set.
File 'default.cfg' was deleted.
The configuration was saved into file 'default.cfg'.
```

Figure 5-112 Saving the configuration file

16. A window opens. In the Filename field, either select an existing configuration file (which will be overwritten) or enter a new file name.

> **Note:** The default.cfg file name is the default name for Load Balancer. When you start the Dispatcher server (dsserver), it looks for the default.cfg file, and if it exists, it loads it. The default.cfg file is stored in *LB_install_path*/servers/configurations/dispatcher.

17. Check if the following directive exists in the generated configuration file. Replace *clusterlb* with your dispatcher's cluster address.

    ```
    dscontrol executor configure clusterlb
    ```

    If the directive does not exist, then add it toward the bottom of the file after the definitions of the servers in the cluster.

The basic configuration is now complete.

5.6.2 Configuring the Load Balancer: Primary and backup scenario

This scenario presents the Load Balancer configured on two servers: primary and backup. These servers load balance the traffic between two Web servers as shown in Figure 5-113.

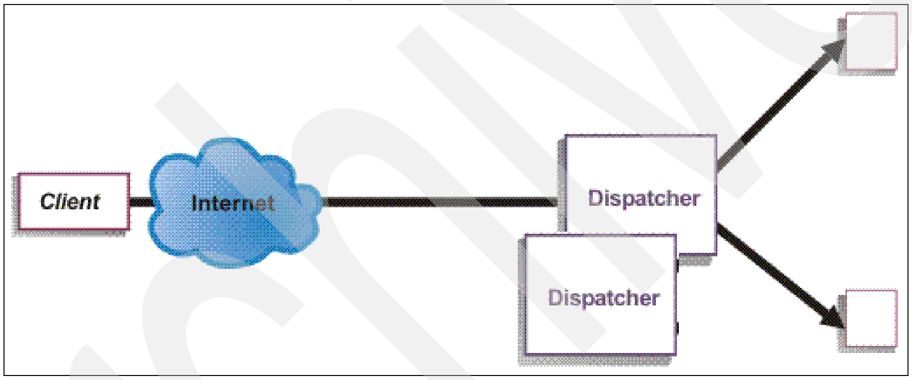


Figure 5-113 Load Balancer primary and backup scenario

We recommend that first you set up the basic scenario and test the load balancing (see 5.6.1, "Configuring the Load Balancer: A simple scenario" on page 172). Then when this is working, use the steps in the following sections to add the second Load Balancer server and the high availability configuration.

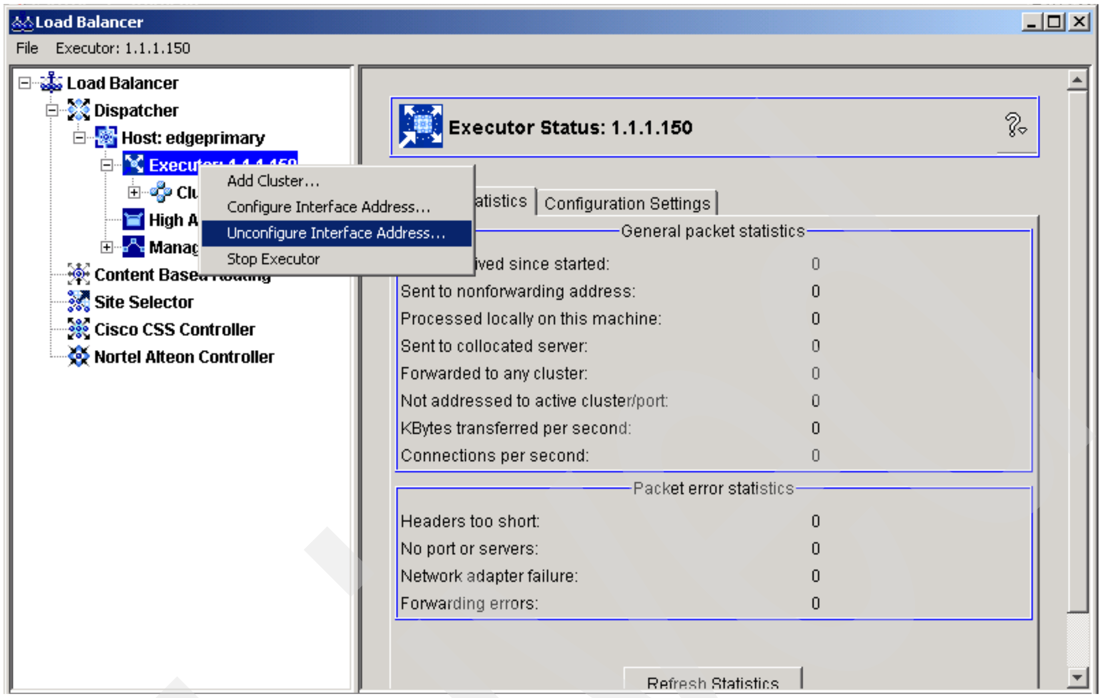
> **Note:** We can use the configuration file that was created for the basic scenario as a starting point for this scenario.

Configuring the Load Balancer Primary server

Now we configure the Load Balancer Primary server in our high availability scenario. First we remove the cluster IP alias from the existing basic configuration.

1. Open the Load Balancer GUI and connect to the primary server as explained in step 1 on page 172 through step 5 on page 174. Make sure that the basic configuration is loaded by checking that the cluster, port, and servers are already configured.

2. Right-click **Executor: 1.1.1.150** and select **Unconfigure Interface Address**, as shown in Figure 5-114.



General packet statistics

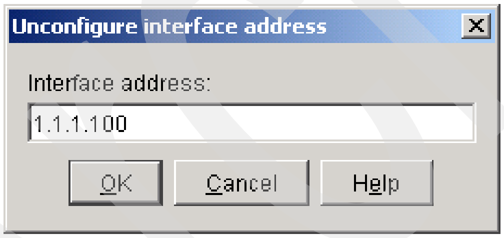
ived since started:	0
Sent to nonforwarding address:	0
Processed locally on this machine:	0
Sent to collocated server:	0
Forwarded to any cluster:	0
Not addressed to active cluster/port:	0
KBytes transferred per second:	0
Connections per second:	0

Packet error statistics

Headers too short:	0
No port or servers:	0
Network adapter failure:	0
Forwarding errors:	0

Figure 5-114 Selecting to unconfigure the interface address

3. The Unconfigured interface address window (Figure 5-115) opens. In the Interface address field, type the IP address. In this scenario, we want to remove the IP alias for the cluster address 1.1.1.100. Click **OK**.



Interface address: 1.1.1.100

Figure 5-115 Unconfigure interface address window

4. Save the current configuration.

5. Copy the configuration to the backup Dispatcher server (second Windows workstation in our case). By doing so, we do not need to set up the basic load balancing configuration there again. Copy this file to the *LB_install_path*/servers/configurations/dispatcher directory.

6. Add the high availability configuration.

   a. Right-click **High Availability** in the left pane of the GUI window and select **Add High Availability Backup**, as shown in Figure 5-116.

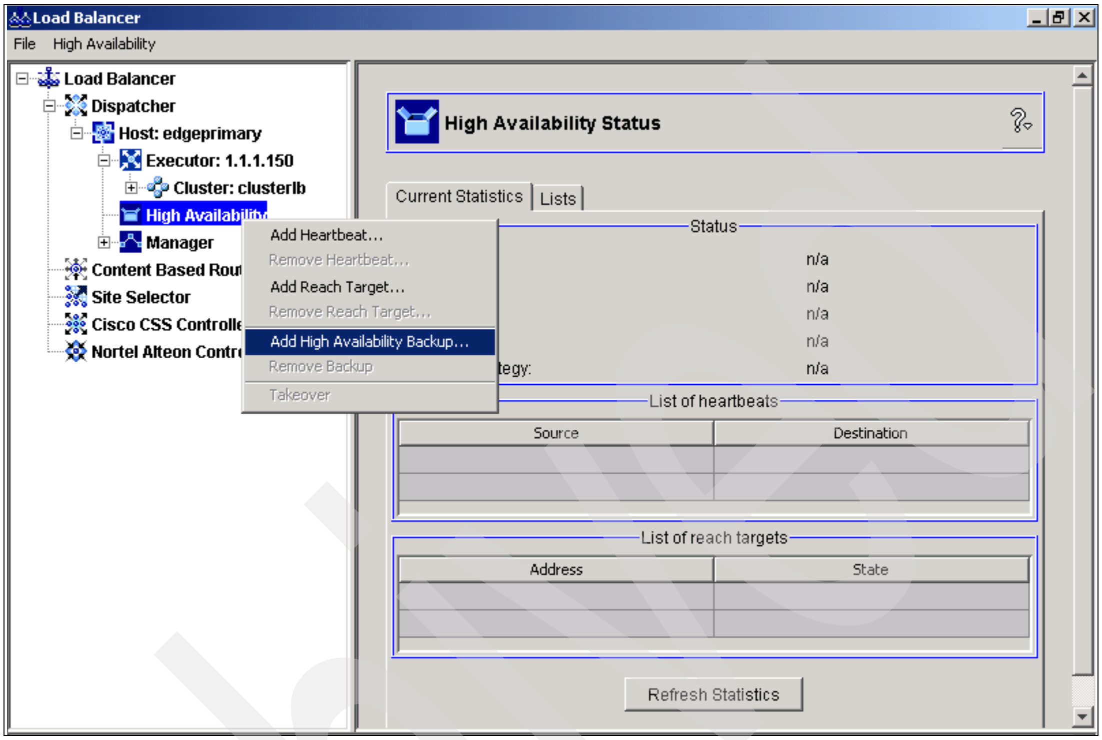


Figure 5-116 Choosing to add a high availability backup

   b. The Configure high availability window (Figure 5-117) opens, in which you complete the following steps.

      i. In the Port number field, type a port number that will be used by both Dispatcher servers to exchange the information needed to keep them synchronized. For our scenario, we chose port 12345.

> **Note:** We can choose any port, but we must ensure that the port number matches on both servers.

      ii. In the Role field, select the role that this server will have in the high availability scenario (Primary, Backup, or Both). In our scenario, this server is our primary server, so we selected **Primary**.

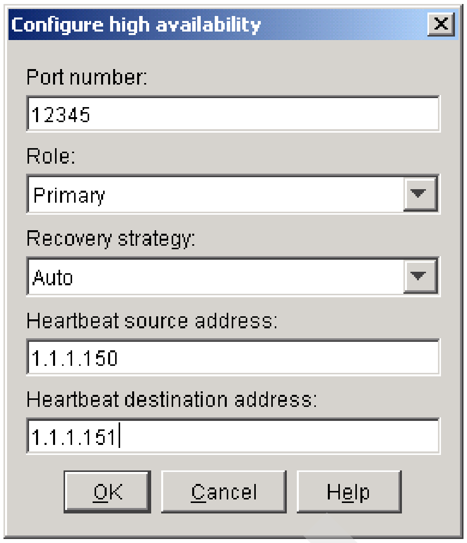
      iii. In the Recovery strategy field, choose how your primary server is going to behave in case the backup server takes over. If you select Auto, the primary machine resumes routing packets as soon as it becomes operational again. We selected **Auto**.

      iv. In the Heartbeat source address field, type the IP address of the local server.

> **Note:** The heartbeat is a Generic Route Encapsulation (GRE) packet that is sent from the local server to the other server in the same high availability cluster to make sure that it is responding.

v. In the Heartbeat destination address field, type the IP address of the backup server.

vi. Click **OK**.



Configure high availability

Port number: 12345

Role: Primary

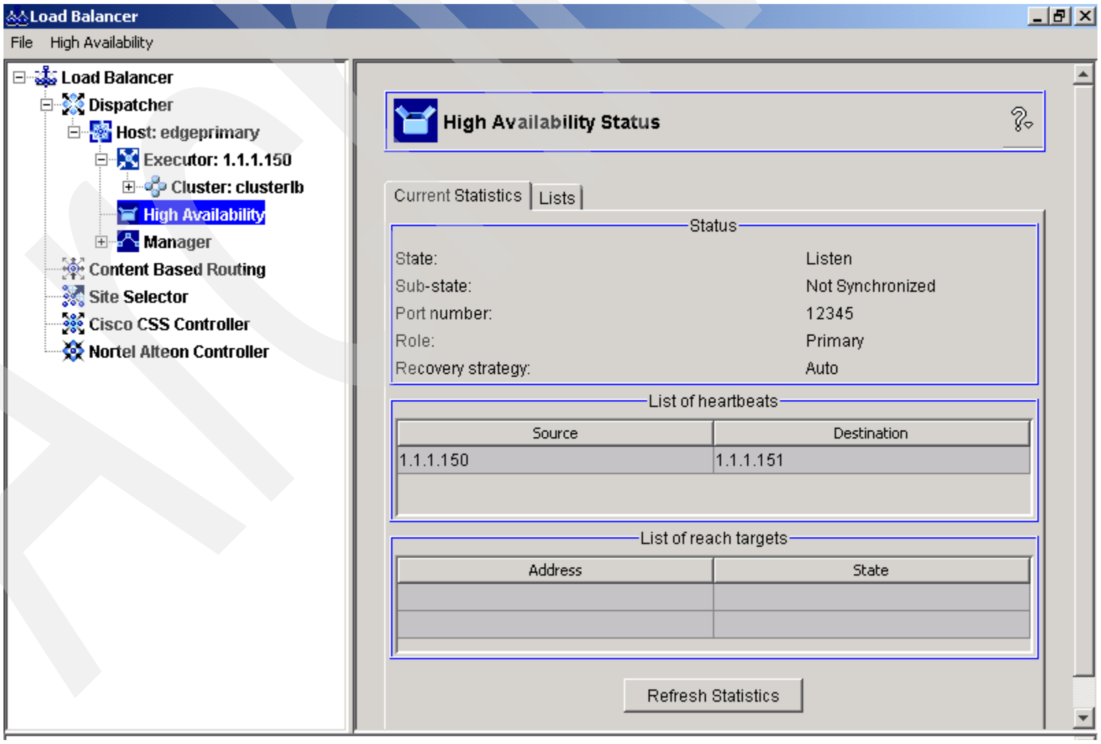
Recovery strategy: Auto

Heartbeat source address: 1.1.1.150

Heartbeat destination address: 1.1.1.151

Figure 5-117 Configure high availability window

7. To see the result of the configuration process on the Edge Primary server, click **High Availability** in the left pane. You can see the high availability status information in the right pane (see Figure 5-118).



Status

State: Listen

Sub-state: Not Synchronized

Port number: 12345

Role: Primary

Recovery strategy: Auto

List of heartbeats

Source	Destination
1.1.1.150	1.1.1.151

List of reach targets

Address	State

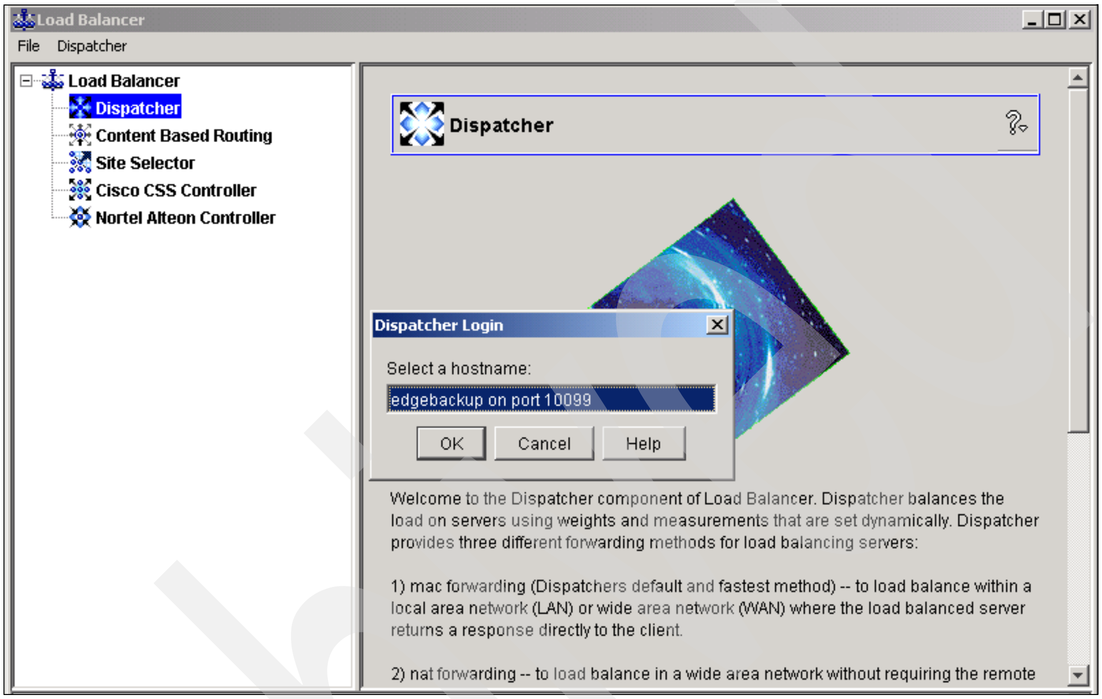
Figure 5-118 High availability status window

You have completed the high availability configuration for the Edge Primary server. Save the configuration as it exists up to this point.

Configuring the Load Balancer backup server

Now we configure the Load Balancer backup server in the high availability scenario.

1. Open the Load Balancer GUI on the second server and connect to the backup server in the same was as you did for the primary server (step 1 on page 172 through step 5 on page 174). In our scenario, the backup server is *edgebackup* (see Figure 5-119).



Select a hostname: edgebackup on port 10099

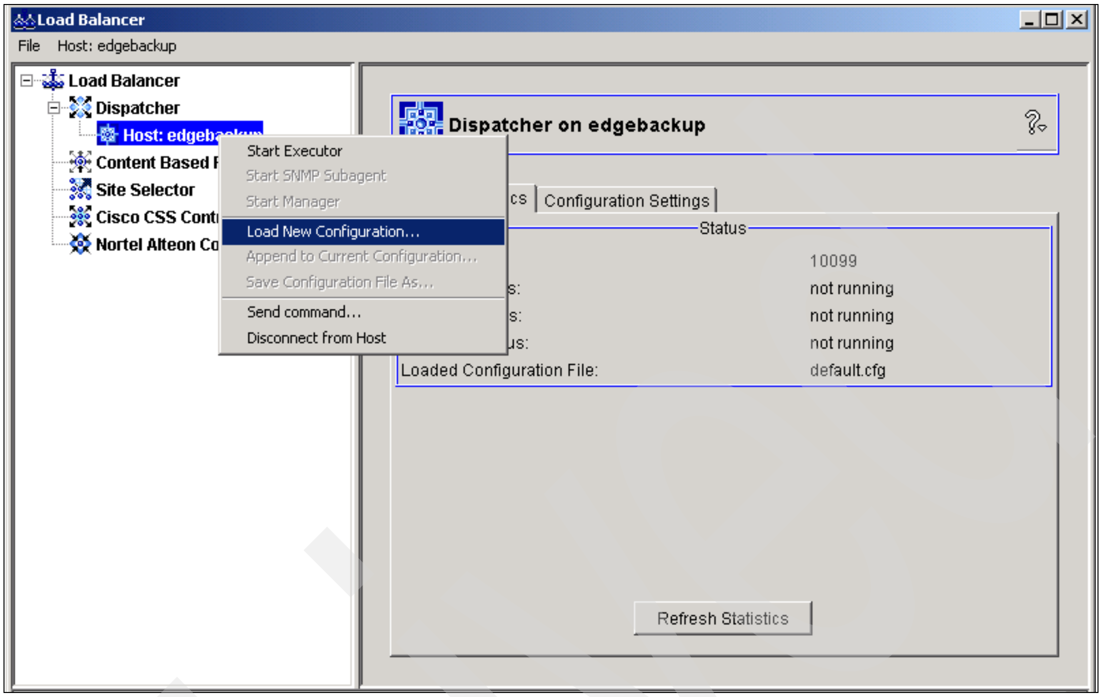
Welcome to the Dispatcher component of Load Balancer. Dispatcher balances the load on servers using weights and measurements that are set dynamically. Dispatcher provides three different forwarding methods for load balancing servers:

1) mac forwarding (Dispatchers default and fastest method) -- to load balance within a local area network (LAN) or wide area network (WAN) where the load balanced server returns a response directly to the client.

2) nat forwarding -- to load balance in a wide area network without requiring the remote

Figure 5-119 Connecting with the Edge Backup server

2. Load the saved configuration file (step 4 on page 184). As shown in Figure 5-120, right-click **Host:edgebackup** and select **Load New Configuration**.

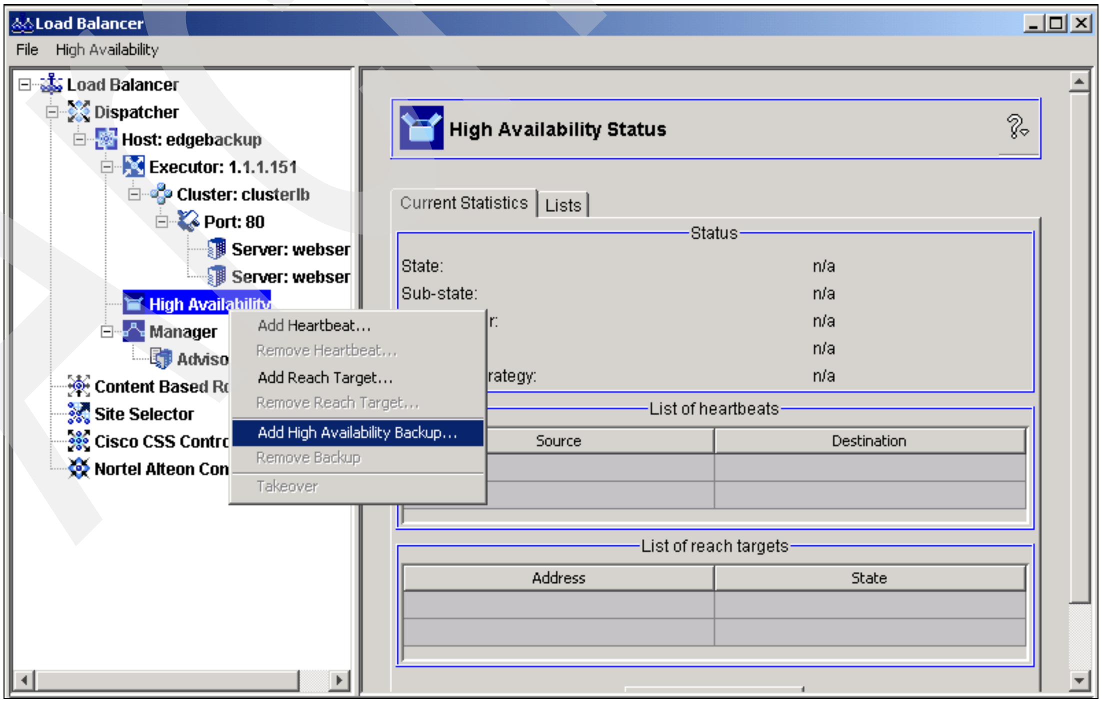


	10099
s:	not running
s:	not running
us:	not running
Loaded Configuration File:	default.cfg

Figure 5-120 Selecting to load the new configuration on the Edge Backup server

3. In the new window that opens, select the saved configuration file and click **OK**.

4. Add the high availability information for the backup server. As shown in Figure 5-121, right-click **High Availability** and select **Add High Availability Backup**.



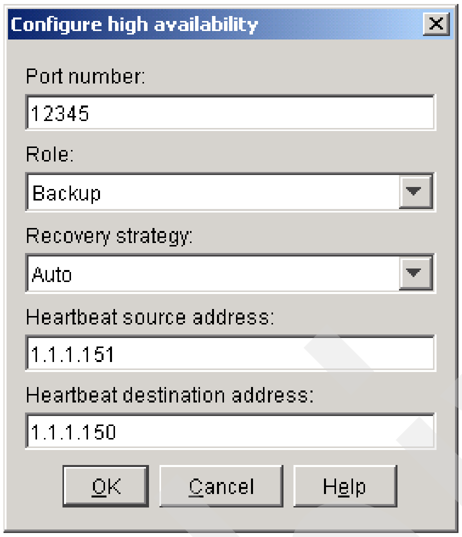
State:	n/a
Sub-state:	n/a
r:	n/a
	n/a
rategy:	n/a

Source	Destination

Address	State

Figure 5-121 Selecting to add a high availability backup

5. The Configure high availability window (Figure 5-122) opens, on which you complete the following steps.
   a. For the Port number and Recovery strategy fields, type the same values that you used in the configuration of the Edge Primary server (see step 6 on page 185).
   b. For the Edge Backup server, the Heartbeat source address is the Edge Backup server itself, and the Heartbeat destination address is the Edge Primary server.
   c. Click **OK**.


Configure high availability

Port number: 12345

Role: Backup

Recovery strategy: Auto

Heartbeat source address: 1.1.1.151

Heartbeat destination address: 1.1.1.150

Figure 5-122 Configuring the high availability Edge Backup server

6. Save your configuration.

5.6.3 Scripts for high availability

For Dispatcher to route packets, each cluster address requires an alias to a network interface device:

- In a stand-alone Dispatcher configuration, create an alias for each cluster address to a network interface card (NIC), for example en0, tr0.
- In a high availability configuration:
  - On the active machine, create an alias for each cluster address to a NIC, for example en0, tr0.
  - On the standby machine, create an alias for each cluster address to a loopback device, for example lo0.

  > **Note:** For Windows systems, create an alias for the cluster address only to the loopback device if you are using the MAC forwarding method with collocated servers.

- In any machine in which the executor has been stopped, remove all aliases to prevent conflicts with another machine that may be started.

Since the Dispatcher machines change states when a failure is detected, it runs user-created scripts to run the commands automatically. You can find sample scripts in the *LB_install_path*/servers/samples directory. Move them to the *LB_install_path*/servers/bin directory to run them. The scripts run automatically only if `dsserver` is running.

For more information, see the *Load Balancer Administration Guide*, GC31-6858, at:

http://www-306.ibm.com/software/webservers/appserv/doc/v602/ec/infocenter/edge/LBguide.htm#HDRSCRUSE

Or refer to the *WebSphere Application Server V6 Scalability and Performance Handbook,* SG24-6392.

Here are some specific details about creating the scripts:

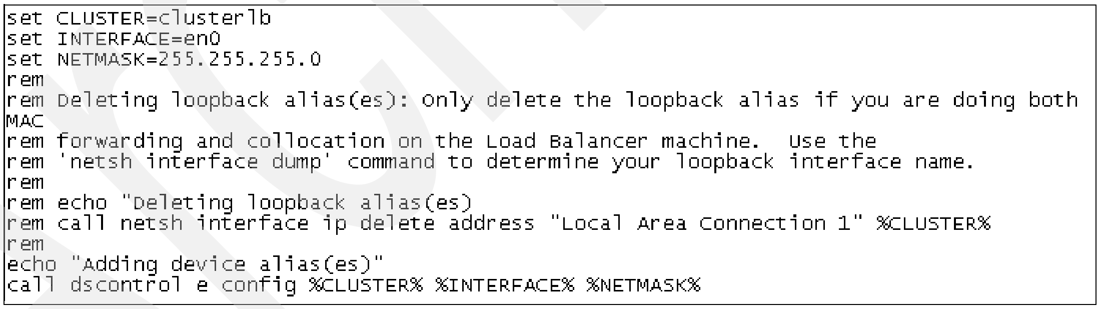
- Load Balancer is case sensitive.
- The scripts are identical for the primary and the backup Dispatcher servers, unless there is a particular command that you need to run on each machine.

For the high availability scenario, you must create at least the following scripts:

- **goActive**: The goActive script executes when Dispatcher goes into active state and begins routing packets.
- **goStandby**: The goStandby script executes when Dispatcher goes into standby state monitoring the health of the active machine, but not routing any packets.
- **goInOp**: The goInOp script executes when a Dispatcher executor is stopped.
- **goIdle** (only needed for a *Windows* workstation): The goIdle script executes when Dispatcher goes into an idle state and begins routing packets. This occurs when the high availability features have not been added, as in a stand-alone configuration. It also occurs in a high availability configuration before the high availability features have been added or after they have been removed.

goActive script

The goActive script executes when Dispatcher goes into active state and begins routing packets. This script deletes loopback aliases and adds device aliases. Figure 5-123 shows an example of the goActive script.



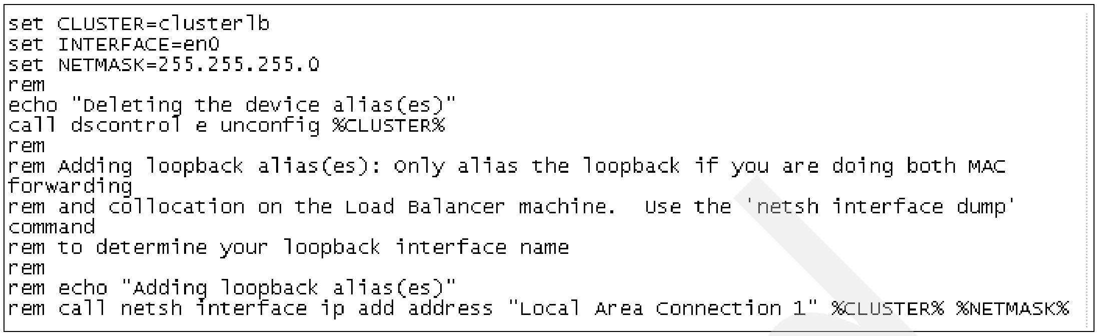
```
set CLUSTER=clusterlb
set INTERFACE=en0
set NETMASK=255.255.255.0
rem
rem Deleting loopback alias(es): Only delete the loopback alias if you are doing both
MAC
rem forwarding and collocation on the Load Balancer machine.  Use the
rem 'netsh interface dump' command to determine your loopback interface name.
rem
rem echo "Deleting loopback alias(es)
rem call netsh interface ip delete address "Local Area Connection 1" %CLUSTER%
rem
echo "Adding device alias(es)"
call dscontrol e config %CLUSTER% %INTERFACE% %NETMASK%
```

Figure 5-123 The goActive script for a high availability scenario

Note: In a basic scenario, you do not need this script.

goStandby script

The goStandby script executes when Dispatcher goes into standby state, monitoring the health of the active machine, but not routing any packets. This script should delete device aliases and add loopback aliases. Figure 5-124 shows an example of the goStandby script.



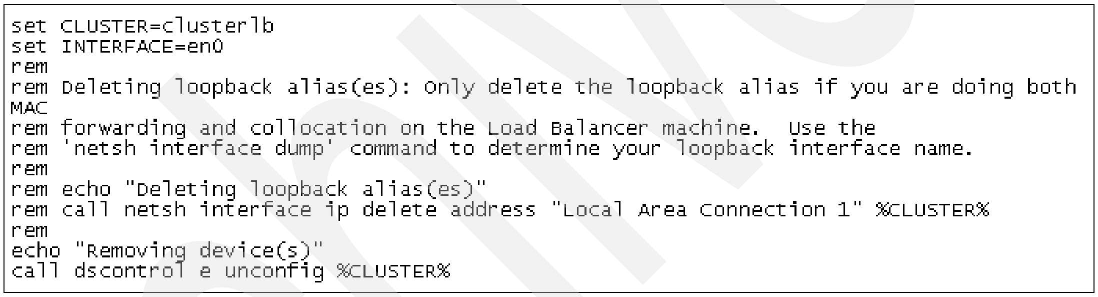
```
set CLUSTER=clusterlb
set INTERFACE=en0
set NETMASK=255.255.255.0
rem
echo "Deleting the device alias(es)"
call dscontrol e unconfig %CLUSTER%
rem
rem Adding loopback alias(es): Only alias the loopback if you are doing both MAC
forwarding
rem and collocation on the Load Balancer machine.  Use the 'netsh interface dump'
command
rem to determine your loopback interface name
rem
rem echo "Adding loopback alias(es)"
rem call netsh interface ip add address "Local Area Connection 1" %CLUSTER% %NETMASK%
```

Figure 5-124 The goStandby script for a high availability scenario

> **Note:** In a basic scenario, we do not need this script.

goInOp script

The goInOp script executes when a Dispatcher executor is stopped and before it is started for the first time. This script deletes all devices and loopback aliases. Figure 5-125 shows an example of the goInOp script.



```
set CLUSTER=clusterlb
set INTERFACE=en0
rem
rem Deleting loopback alias(es): Only delete the loopback alias if you are doing both
MAC
rem forwarding and collocation on the Load Balancer machine.  Use the
rem 'netsh interface dump' command to determine your loopback interface name.
rem
rem echo "Deleting loopback alias(es)"
rem call netsh interface ip delete address "Local Area Connection 1" %CLUSTER%
rem
echo "Removing device(s)"
call dscontrol e unconfig %CLUSTER%
```

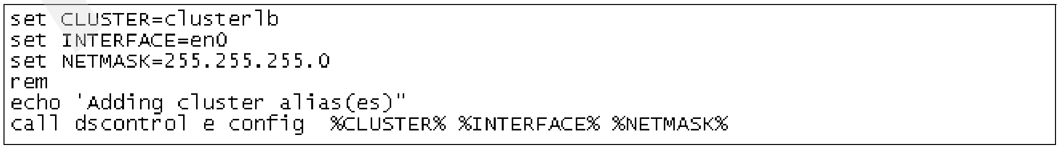
Figure 5-125 The goInOp script for a high availability scenario

> **Note:** In a basic scenario, this script is optional. We may create it and have it delete device aliases, or we may choose to delete them manually.

goIdle script

The goIdle script executes when Dispatcher goes into the idle state and begins routing packets. This occurs when the high availability features have not been added, as in a basic scenario. It also occurs in a high availability configuration before the high availability features have been added or after they have been removed. However, for Windows systems running high availability, you need this script.

Figure 5-126 shows an example of the goIdle script.



```
set CLUSTER=clusterlb
set INTERFACE=en0
set NETMASK=255.255.255.0
rem
echo 'Adding cluster alias(es)"
call dscontrol e config  %CLUSTER% %INTERFACE% %NETMASK%
```

Figure 5-126 The goIdle script for a high availability scenario

Notes:

- When normally running Dispatcher in a high availability scenario, we should not create this script, but for Windows systems running high availability, we need this script.
- For Dispatcher in a basic scenario, this script is optional.

5.6.4 Testing the Load Balancer scenario

You should test the Load Balancer cluster to see if the configuration works. You can do this by using five methods.

First method

From a Web browser, go to the location `http://clusterlb`. If a page appears, Load Balancer is operational.

Shut down the edgeprimary machine and reload the page in the Web browser. If a page appears (see Figure 5-127), the switchover works well.

Important: Make sure you turn off the caching feature in your browser.

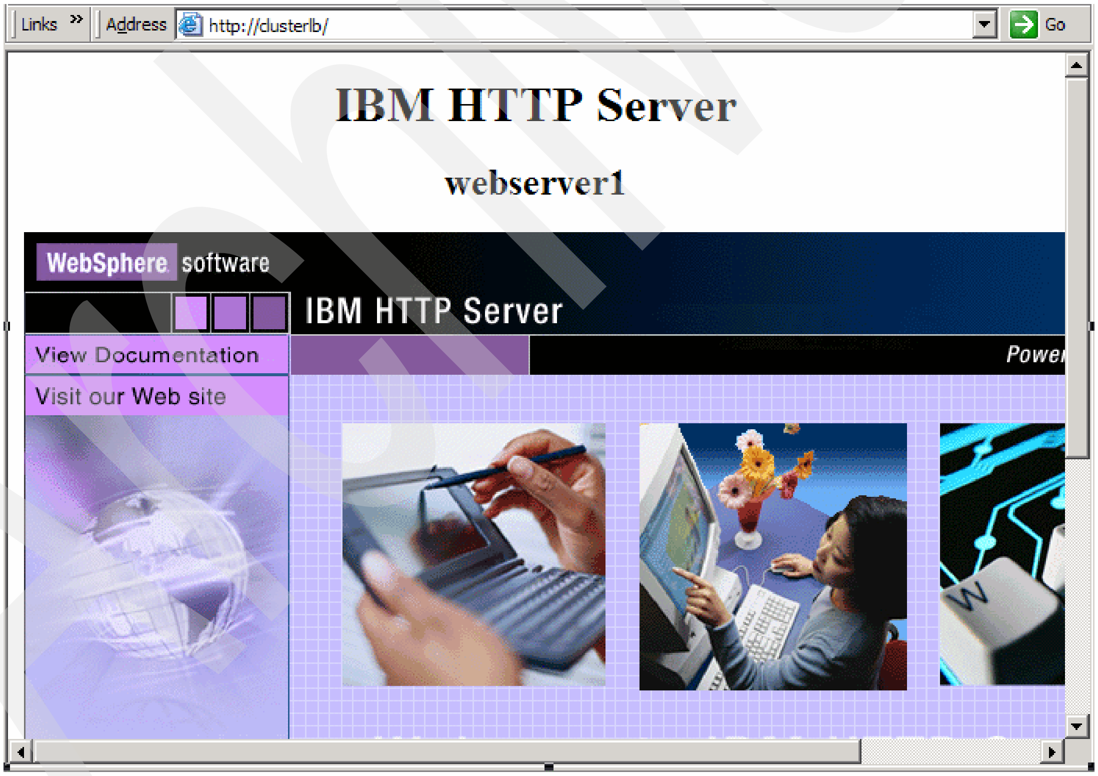


Figure 5-127 Loading http://clusterlb/

Now start the edgeprimary machine, shut down the edgebackup machine, and reload the page in the Web browser. If a page appears, it's working well.

Second method

Dispatcher provides reports that you can use to verify the configuration. You can see whether the back-end servers that make up the cluster are active and sending responses to the advisors. You can also see if the traffic is being balanced. Use the server monitor on the GUI (see Figure 5-128).

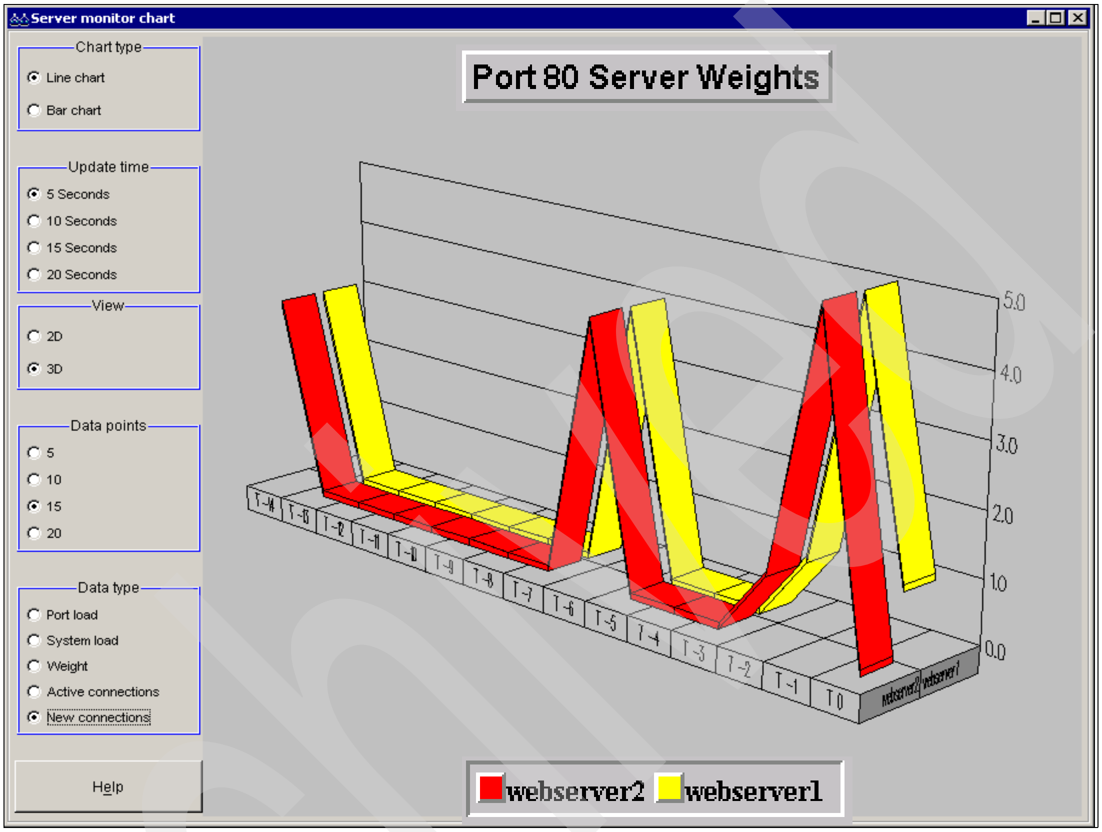


Chart type
- ● Line chart
- ○ Bar chart

Update time
- ● 5 Seconds
- ○ 10 Seconds
- ○ 15 Seconds
- ○ 20 Seconds

View
- ○ 2D
- ● 3D

Data points
- ○ 5
- ○ 10
- ● 15
- ○ 20

Data type
- ○ Port load
- ○ System load
- ○ Weight
- ○ Active connections
- ● New connections

🟥 webserver2 🟨 webserver1

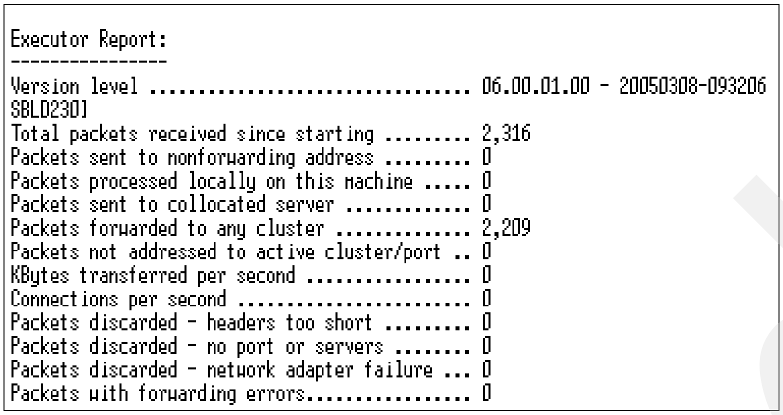
Figure 5-128 Monitoring administration tools: New connections

Third method

You can check whether packets are being forwarded to the cluster by typing the following command in the command window of the Load Balancer workstation:

```
dscontrol executor report
```

This command produces a report of the packet traffic on the Executor component of Dispatcher (see Figure 5-129).



```
Executor Report:
----------------
Version level ................................. 06.00.01.00 - 20050308-093206
SBLD2301
Total packets received since starting ......... 2,316
Packets sent to nonforwarding address ......... 0
Packets processed locally on this machine ..... 0
Packets sent to collocated server ............. 0
Packets forwarded to any cluster .............. 2,209
Packets not addressed to active cluster/port .. 0
KBytes transferred per second ................. 0
Connections per second ........................ 0
Packets discarded - headers too short ......... 0
Packets discarded - no port or servers ........ 0
Packets discarded - network adapter failure ... 0
Packets with forwarding errors................. 0
```

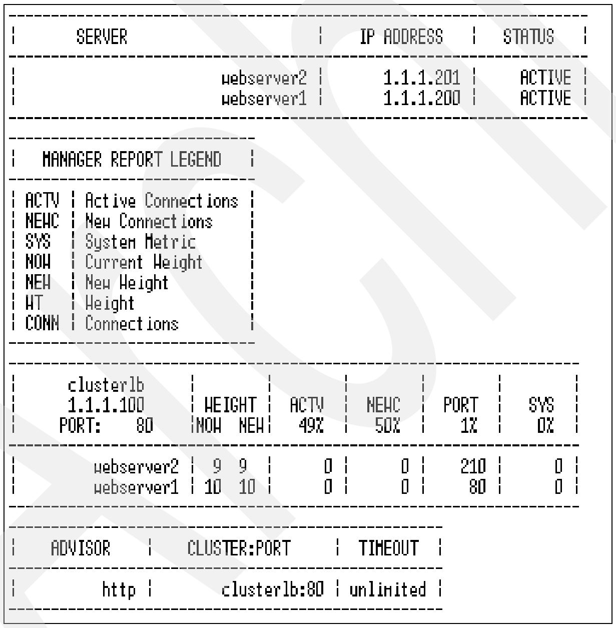
Figure 5-129 Result of running the dscontrol executor report command

Fourth method

Type the following command on the Load Balancer workstation:

```
dscontrol manager report
```

As shown in the example in Figure 5-130, the first table lists the back-end servers being load balanced and their status. The second table lists the servers by port, weight, number of active and new connections, and load values. The last table shows the advisors that were started, the port, and the time-out value attributed to it.



```
--------------------------------------------------------------------
|      SERVER                  |   IP ADDRESS   |   STATUS   |
--------------------------------------------------------------------
|                   webserver2 |      1.1.1.201 |     ACTIVE |
|                   webserver1 |      1.1.1.200 |     ACTIVE |
--------------------------------------------------------------------
-----------------------------
|   MANAGER REPORT LEGEND   |
-----------------------------
| ACTV | Active Connections |
| NEWC | New Connections    |
| SYS  | System Metric      |
| NOW  | Current Weight     |
| NEW  | New Weight         |
| WT   | Weight             |
| CONN | Connections        |
-----------------------------
-------------------------------------------------------------------
|     clusterlb      |         |        |        |        |        |
|     1.1.1.100      | WEIGHT  |  ACTV  |  NEWC  |  PORT  |   SYS  |
|    PORT:    80     |NOW  NEW |   49%  |   50%  |    1%  |    0%  |
-------------------------------------------------------------------
|         webserver2 |  9   9  |      0 |      0 |    210 |      0 |
|         webserver1 | 10  10  |      0 |      0 |     80 |      0 |
-------------------------------------------------------------------
---------------------------------------------------
|   ADVISOR    |    CLUSTER:PORT     |  TIMEOUT  |
---------------------------------------------------
|         http |        clusterlb:80 | unlimited |
---------------------------------------------------
```

Figure 5-130 Result of running the dscontrol manager report command

Fifth method

Use one of the stress test tools, such as Rational Performance Tester, to generate the workload for your Web servers through the Load Balancer. Use either the Load Balancer monitor or traffic report to see if Load Balancer works correctly.

5.7 Configuring an LDAP server

This section presents the basic steps to help you configure Directory Server for iSeries to secure your WebSphere Application Server environment. It also provides guidance about a possible high availability configuration for the LDAP server.

Directory Server for iSeries provides an LDAP server on the iSeries server. LDAP runs over TCP/IP and is popular as a directory service for both Internet and non-Internet applications.

Directory Server for iSeries has enhancements and new features for V5R3. The new Directory Server Web administration tool replaces the IBM Directory Management Tool (DMT). The Web administration tool includes the functionality to administer user entries, directory server processes, and the directory tree from one common Web interface.

5.7.1 Directory server concepts

There are some basic concepts that you should understand in regard to the configuration steps that you will follow to set up your LDAP server.

Directories

The Directory Server allows access to a type of database that stores information in a hierarchical structure similar to the way that the OS/400 IFS is organized.

If the name of an object is known, its characteristics can be retrieved. If the name of a particular individual object is not known, the directory can be searched for a list of objects that meet a certain requirement. Directories can usually be searched by either specific criteria or a predefined set of categories.

A *directory* is a specialized database that has characteristics that set it apart from general purpose relational databases. A characteristic of a directory is that it is accessed (read or searched) much more often than it is updated (written). Because directories must be able to support high volumes of read requests, they are typically optimized for read access. Because directories are not intended to provide as many functions as general-purpose databases, they can be optimized to economically provide more applications with rapid access to directory data in large distributed environments.

A directory can be centralized or distributed. If a directory is centralized, one directory server (or a server cluster) at one location provides access to the directory. If the directory is distributed, multiple servers, usually geographically dispersed, provide access to the directory.

When a directory is distributed, the information stored in the directory can be partitioned or replicated. When information is partitioned, each directory server stores a unique and non-overlapping subset of the information. That is, each directory entry is stored by one and only one server. The technique to partition the directory is to use *LDAP referrals*. LDAP referrals allow users to refer LDAP requests to either the same or different name spaces stored in a different (or same) server. When information is replicated, the same directory entry is stored by more than one server. In a distributed directory, some information may be partitioned, and some information may be replicated.

The LDAP directory server model is based on entries, which are also referred to as *objects*. Each entry consists of one or more attributes, such as a name or address, and a type. The type typically consists of mnemonic strings, such as cn for common name, mail for e-mail address, fax, title, sn (for surname), or jpegPhoto.

Each directory has a schema, which is a set of rules that determine the structure and contents of the directory.

Each directory entry has a special attribute called an *objectClass*. This attribute controls which attributes are required and allowed in an entry. The values of the objectClass attribute determine the schema rules that the entry must obey.

Distinguished names

Every entry in the directory has a distinguished name (DN). The DN uniquely identifies an entry in the directory and is made up of attribute-value pairs, separated by commas.

Any of the attributes defined in the directory schema may be used to make up a DN. The order of the component attribute value pairs is important. The DN contains one component for each level of the directory hierarchy from the root down to the level where the entry resides. LDAP DNs begin with the most specific attribute, usually a sort of name, and continue with progressively broader attributes, often ending with a country attribute. The first component of the DN is referred to as the Relative Distinguished Name (RDN™). It identifies an entry distinctly from any other entries that have the same parent.

Suffix (naming context)

A suffix, also known as a *naming context*, is a DN that identifies the top entry in a locally held directory hierarchy. Because of the relative naming scheme used in LDAP, this DN is also the suffix of every other entry within that directory hierarchy. A directory server can have multiple suffixes, each identifying a locally held directory hierarchy, for example:

```
o=ibm,c=us
```

The specific entry that matches the suffix must be added to the directory. The entry that you create must use an objectClass that contains the naming attribute used. You can use the Web administration tool or the Qshell **ldapadd** utility to create the entry that corresponds to this suffix.

There are two commonly used naming conventions for suffixes. One is based on the TCP/IP domain for your organization. The other is based on your organization's name and location.

Realms and user templates

The realm and template objects found in the Web administration tool are used to relieve the user of the need to understand some of the underlying LDAP issues.

A *realm* identifies a collection of users and groups. It specifies information, in a flat directory structure, such as where users are located and where groups are located. A realm defines a location for users, for example "cn=users,o=acme,c=us", and creates users as immediate subordinates of that entry. For example, John Doe is created as "cn=John Doe,cn=users,o=acme,c=us". You can define multiple realms and give them familiar names, such as “Web Users”. The familiar name can be used by the people who are creating and maintaining the users.

A *template* describes what a user looks like. It specifies the objectClasses that are used when creating users (both the structural objectClass and any auxiliary classes that you want). A template also specifies the layout of the panels used to create or edit users (for example, names of tabs, default values, and attributes to appear on each tab).

When you add a new realm, you create an ibm-realm object in the directory. The ibm-realm object keeps track of the properties of the realm such as where users and groups are defined, and what template to use. The ibm-realm object can point to an existing directory entry that is the parent of users, or it can point to itself (the default), making it the container for new users.

LDAP directory referrals

Referrals allow Directory Servers to work in teams. If the DN that a client requests is not in one directory, the server can automatically send (refer) the request to any other LDAP server.

Directory Server allows you to use two different types of referrals. You can specify default referral servers, where the LDAP server will refer clients whenever any DN is not in the directory. You can also use your LDAP client to add entries to the directory server that have the objectClass referral. This allows you to specify referrals that are based on the specific DN that a client requests.

Referral servers are closely related to replica servers. Because data on replica servers cannot be changed from clients, the replica refers any requests to change directory data to the master server.

5.7.2 Setting up the Directory Server for iSeries

The configuration of the Directory Server requires you to perform the following tasks:

1. Set up the Directory Server
2. Set up the Web administration tool
3. Create the directory database
4. Enter information into the directory database
5. Test the directory database

Setting up the Directory Server

Note: You must have *ALLOBJ and *IOSYSCFG special authorities to configure the server.

Perform the following steps:

1. In iSeries Navigator, expand ***your system*** → **Network** → **Servers** and click **TCP/IP**.
2. In the Server Configuration tasks pane, at the bottom of iSeries Navigator window, click **Configure system as Directory server**.
3. The Directory Server Configuration Wizard appears.

   If the wizard does not appear, right-click **IBM Directory Server** in the list of TCP/IP and select **Stop**. Right-click it again and select **Reconfigure**.
4. In the IBM Directory Server Configuration Wizard - Welcome panel, select **Configure a local LDAP directory server**, and click **Next**.
5. In the IBM Directory Server Configuration Wizard - Specify Settings panel, select **No**, so you can configure the LDAP server without the default settings. Then click **Next**.
6. In the IBM Directory Server Configuration Wizard - Specify Administrator DN panel, deselect **System-generated**. Then enter the data as shown in Table 5-11.

   Table 5-11 Administrator DN

Field	Value
Administrator DN	cn=administrator
Password	Password of your choice
Confirm password	Re-type the password

   Click **Next**.

7. In the IBM Directory Server Configuration Wizard - Specify Suffixes panel, complete the following tasks:

   a. In the Suffix field, type `dc=<your company>,dc=com`.
   b. Click **Add** to add a new root for your data.
   c. Click **Next**.

8. In the IBM Directory Server Configuration Wizard - Select IP Addresses panel, select **Yes, use all IP addresses**, and click **Next**.

9. In the IBM Directory Server Configuration Wizard - Specify TCP/IP Preference panel, select **Yes**, and click **Next**.

10. In the IBM Directory Server Configuration Wizard - System Information to Publish, click **Next**.

11. In the IBM Directory Server Configuration Wizard - Summary panel, click **Finish**.

12. In the list of TCP/IP servers, right-click **IBM Directory Server** and select **Start**.

Setting up the Web administration tool

The initial setup for the Web administration tool includes two main steps: creating a WebSphere instance and configuring the tool itself.

Setting up Web administration for the first time

Complete the following steps to set up the Directory Server Web Administration Tool for the first time.

1. Install IBM WebSphere Application Server - Express V5.0 (5722-IWE Base and Option 2) and the associated prerequisite software if they are not already installed.

   For more information, look under the path **iSeries Information Center, Version 5 Release 3 → e-business and Web serving → Application servers → WebSphere Application Server - Express V5 → Installation → Step 1: Plan the installation and initial configuration** in the iSeries Information Center at:

   `http://publib.boulder.ibm.com/infocenter/iseries/v5r3/index.jsp`

   > **Important:** In V5R3, only WebSphere Application Server - Express V5.0 is supported for the Web Administration tool.

2. Enable the system application server instance through the HTTP ADMIN server instance.

   a. Start the HTTP ADMIN server instance by using one of the following actions:

      - In iSeries Navigator, expand ***your system*** → **Network** → **Servers** and click **TCP/IP**. Right-click **HTTP Administration** from the list of servers and select **Start**.
      - On an OS/400 command line run the following command:

        ```
        STRTCPSVR SERVER(*HTTP) HTTPSVR(*ADMIN)
        ```

   b. Log on to the IBM Web Administration for iSeries. Use an OS/400 user profile and password to logon to the iSeries Tasks page at `http://your_server:2001`.

   c. Click **IBM Web Administration for iSeries**. Complete the following steps in Web Administration for iSeries:

      i. Click the **Manage** tab.
      ii. Click the **HTTP Servers** subtab.
      iii. In the Server list, make sure that **ADMIN - Apache** is selected.

   d. From the options in the left pane of the page, click **General Server Configuration**.

   > **Note:** You might need to expand the Server Properties section to see the General Server Configuration option.

      i. On the General Settings tab, set the Start the system application server instance when the 'Admin' server is started parameter to **Yes**.

      ii. Click **OK**.

3. Restart the HTTP ADMIN server instance by clicking the restart button (the second button under the HTTP Servers tab).

4. Log on to the Directory Server Web Administration Tool.

   a. Access the iSeries Tasks page at `http://`*`your_server`*`:2001`.
   b. Click **IBM Directory Server for iSeries**.
   c. In the LDAP Hostname field, select **Console Admin**.
   d. In the Username field, type `superadmin` (the default user ID.)
   e. In the Password field, type `secret` (the default password).
   f. Click **Login**. The IBM Directory Server Web Administration Tool page is displayed.

5. Change the console administration login.

   a. Click **Console administration** in the left pane to expand the section, and then click **Change console administrator login**.

   b. Type a new console administration login name in the Console administrator login field.

   c. In the Current password field, type the current password (`secret`).

   d. Click **OK**.

6. Change the console administration password by clicking **Change console administrator password** in the left pane.

7. If you have multiple IBM Directory Servers, you can add them to the list of the servers to be managed through the Web Administration tool. Add the Directory Server that you want to administer.

   a. Click **Manage console servers** in the left pane.

   b. Click the **Add** button.

   c. In the new form, type the host name of your LDAP server and change, if necessary, the port numbers.

   d. Click **OK**.

8. Click **Logout**.

9. When the Logout successful page appears, click the **here** link to return to the Web administration login page.

10. From this point onward, you can work with your LDAP server using Web administration tool. You simply select your LDAP server's host name from the LDAP Hostname field and enter your LDAP administrator's DN and password (see Table 5-11 on page 197).

Now that you have the Directory server up and running, you can populate it with information about users and groups. See the iSeries Information Center to learn more about working with the Directory server.

`http://publib.boulder.ibm.com/infocenter/iseries/v5r3/index.jsp?lang=en`

Creating the directory database

Before you can begin to enter data, you must create a place within Directory server for the data to be stored. This task includes several steps.

Login to the Web administration tool.

1. Access the iSeries Tasks page at `http://your_server:2001`. Login using your iSeries credentials.
2. Click **IBM Directory Server for iSeries**.
3. In the login page, from the LDAP Hostname list, select your LDAP server system.
4. Use your LDAP administrator common name (cn) and password. In our example, the administrator name is cn=administrator. This is the information that you set up during the configuration of your LDAP server in step 6 on page 197.
5. Click **Login**.

Step 1: Creating a base DN object

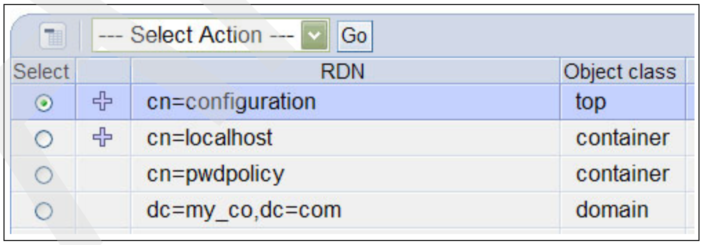
Create a distinguished name under which you will add all the information related to your WebSphere configuration.

1. In the Web administration tool, click **Directory management** → **Manage entries**.
2. You see a listing of the objects in the base level of the directory. Since the server is new, you see only the structural objects which contain the configuration information.

   You want to add a new object to contain the MyCo, Inc. (this name is given for example purposes) data. Click **Add** on the right side of the window.
3. In the next window, scroll within the Object class list and select **domain**. Then click **Next**.
4. In the next window, you do not want to add any auxiliary object classes, so click **Next** again.
5. In the Enter the attributes window, enter the data that corresponds with the suffix that you created earlier in the IBM Directory server wizard:

   a. Leave the Object class list set to **domain**.
   b. In the Relative DN field, type `dc=my_co`.
   c. In the Parent DN field, type `dc=com`.
   d. In the dc field, type `my_co`.
   e. At the bottom of the window, click **Finish**.

Back in the base level, you should see the new base DN (see Figure 5-131).



Select		RDN	Object class
●	✚	cn=configuration	top
○	✚	cn=localhost	container
○		cn=pwdpolicy	container
○		dc=my_co,dc=com	domain

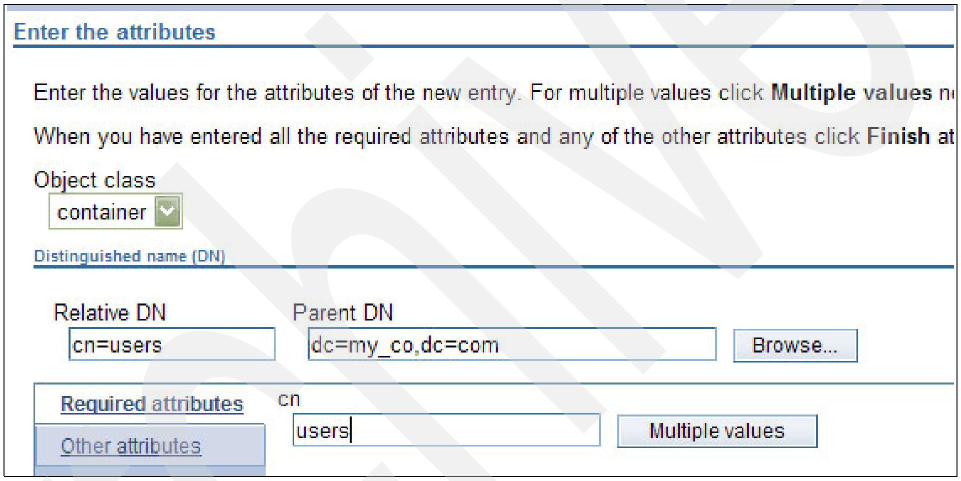
Figure 5-131 Adding a new domain

Next, you use this domain to create and store all information related either to your company or a specific application, like a human resources application. To follow best practices scenario, we create several additional objects of type container to separate all data for better manageability. We create a separate container for:

- **Users**: This container holds user information.
- **Groups**: This container includes all the groups, such as administrative users, power users, and so on. Each group of users has its own set of authorities and other attributes that limit its members to a specific set of actions that they can perform. Typically, each user belongs to one or more groups.
- **Auxiliary objects** (like templates; see the following sections): These types of objects don't contain a company's data, but internal Directory Server information.

Perform these steps to create the three containers:

1. In the Web Administration tool, expand **Directory management** and click **Add an entry**.
2. In the right pane, select **Container** as the object class and click **Next**.
3. In the next panel, you don't need to select an auxiliary object class, so click **Next** again.
4. Under Distinguished name (see Figure 5-132), complete the following tasks:
   a. In the Relative DN field, type `cn=users`.
   b. In the Parent DN field, type `dc=my_co,dc=com`.
   c. In the cn field, type `users`.
   d. Click **Finish**.



Enter the attributes

Enter the values for the attributes of the new entry. For multiple values click **Multiple values** n

When you have entered all the required attributes and any of the other attributes click **Finish** at

Object class: container

Distinguished name (DN)

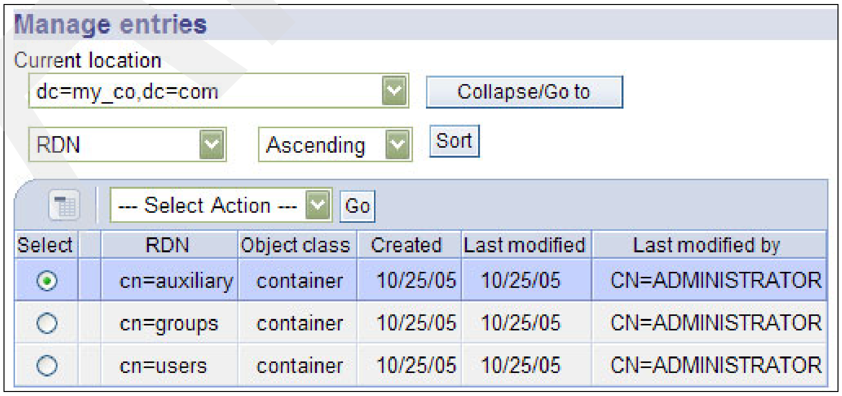
Relative DN: cn=users

Parent DN: dc=my_co,dc=com

cn: users

Figure 5-132 Creating a container

5. Repeat these steps for each of the containers *groups* and *auxiliary*.
6. After you add all three containers, click **Manage entries** in the navigation tree on the left.
7. Select the option **dc=my_co,dc=com** and click **Expand**. You should see three containers listed as shown in Figure 5-133.



Manage entries

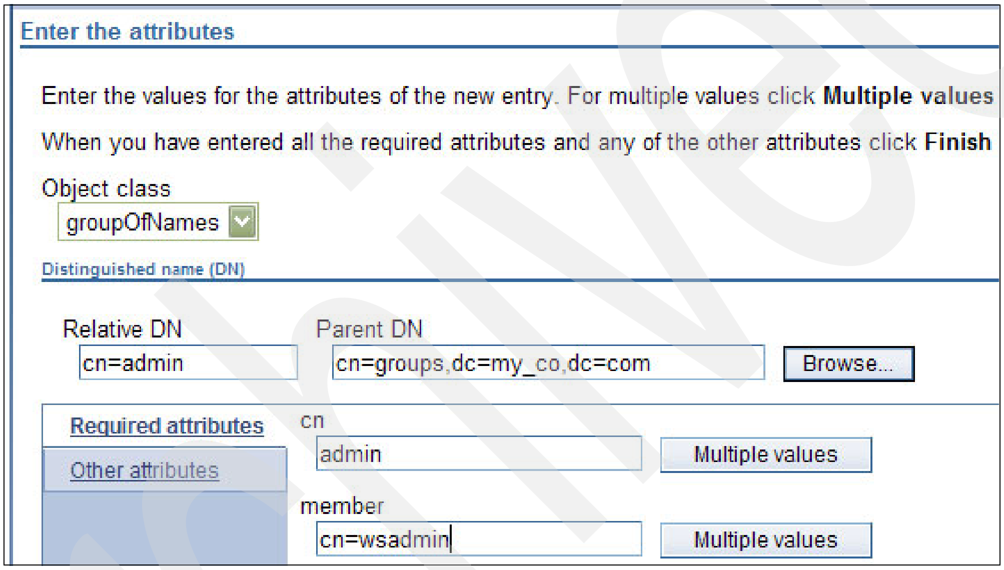
Current location: dc=my_co,dc=com

RDN, Ascending

Select	RDN	Object class	Created	Last modified	Last modified by
●	cn=auxiliary	container	10/25/05	10/25/05	CN=ADMINISTRATOR
○	cn=groups	container	10/25/05	10/25/05	CN=ADMINISTRATOR
○	cn=users	container	10/25/05	10/25/05	CN=ADMINISTRATOR

Figure 5-133 Three containers added

8. Create a group for the administrative users called *admin*. Following this example, you can create an additional group.
9. Select the radio button next to **cn=groups** and click **Add**.
10. In the next panel, from the list of available object classes, select **groupOfNames** and click **Next**.
11. Click **Next** again.
12. In the Enter the attributes panel, under the Distinguished name (DN) section (Figure 5-134), provide the following information:
    a. In the Relative DN field, type `cn=admin`.
    b. In the cn field, type `admin`.
    c. In the member field, type `cn=wsadmin`. This is the first member of the group; you can use your own name.
    d. In the Parent DN field, you should already see the value `cn=groups,dc=my_co,dc=com`.



Enter the attributes

Enter the values for the attributes of the new entry. For multiple values click **Multiple values**

When you have entered all the required attributes and any of the other attributes click **Finish**

Object class: groupOfNames

Distinguished name (DN)

Relative DN: cn=admin

Parent DN: cn=groups,dc=my_co,dc=com

cn: admin

member: cn=wsadmin

Figure 5-134 Creating a group

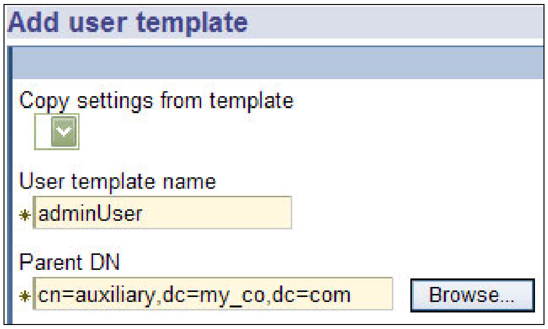
Now, when you need to create a new user, group, or auxiliary LDAP object, you place it in its own container.

Step 2: Creating a user template for administrative users

Next you create a user template as an aid for adding WebSphere administrative users data. You may create a template for different types of users within WebSphere.

1. Click **Realms and templates**.
2. Click **Add user template**.
3. In the Add user template section (Figure 5-135), complete the following tasks:
    a. In the User template name field, type `adminUser`.
    b. For Parent DN, type `cn=auxiliary,dc=my_co,dc=com`.

       Or you can click the **Browse** button next to the Parent DN field, navigate to this container, and click **Select**.
    c. Click **Next**.



Add user template

Copy settings from template

User template name

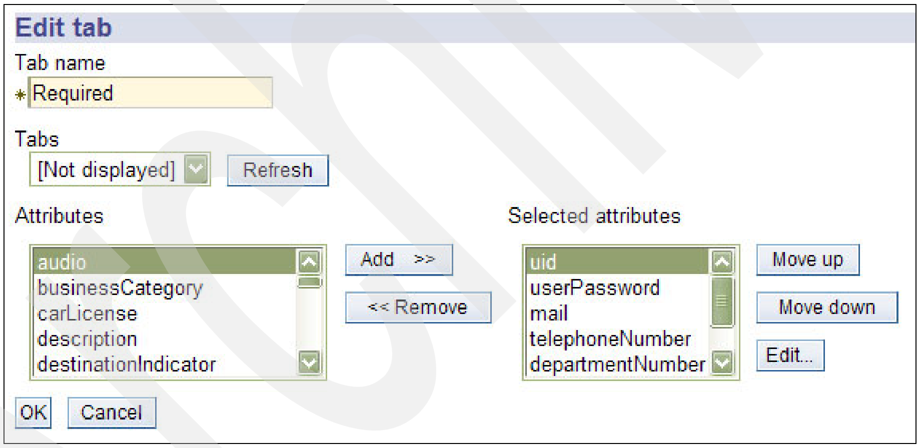
* adminUser

Parent DN

* cn=auxiliary,dc=my_co,dc=com

Figure 5-135 Adding a new template

4. From the Structural object class drop-down list, select **inetOrgPerson**. Click **Next**.
5. From the Naming attribute drop-down list, select **uid**.
6. From the Tabs list, select **Required** and click **Edit**.
7. Under Edit tab, choose which fields to include in the user template. *sn* and *cn* are required. In the Attributes section, complete the following tasks:
   a. Select **departmentNumber** and click **Add**.
   b. Select **telephoneNumber** and click **Add**.
   c. Select **mail** and click **Add**.
   d. Select **userPassword** and click **Add** (see Figure 5-136).
   e. Click **OK**.



Edit tab

Tab name

* Required

Tabs

[Not displayed]

Attributes

audio
businessCategory
carLicense
description
destinationIndicator

Selected attributes

uid
userPassword
mail
telephoneNumber
departmentNumber

Figure 5-136 Adding attributes

8. Click **Finish** to create the user template.

Step 3: Creating a realm

As mentioned in “Realms and user templates” on page 196, a *realm* identifies a collection of users and groups. Now you must create a realm for the administrative users of WebSphere.

1. In the Web Administration tool, click **Realms and templates**.
2. Click **Add realm**.

3. Complete the following tasks:

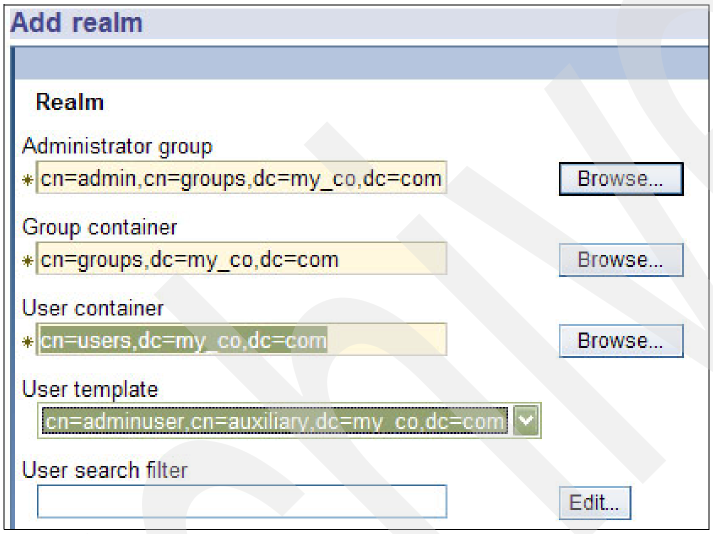
   a. In the Realm name field, type `Administrators`.
   b. For Parent DN, type `cn=auxiliary,dc=my_co,dc=com`.

      Or you can click the **Browse** button next to the Parent DN field, navigate to this container, and click **Select**.

   c. Click **Next**.

4. In the Add realm panel (see Figure 5-137), complete the following actions:

   a. In the Administrator group field, type `cn=admin,cn=groups,dc=my_co,dc=com`.
   b. In the Group container field, type `cn=groups,dc=my_co,dc=com`.
   c. In the User container field, type `cn=users,dc=my_co,dc=com`.
   d. For User template, select **cn=adminuser,cn=auxiliary,dc=my_co,dc=com**. This is the only selection available if you start this example with the empty Directory Server.
   e. Click **Finish**.



Add realm

Realm

Administrator group

* cn=admin,cn=groups,dc=my_co,dc=com

Group container

* cn=groups,dc=my_co,dc=com

User container

* cn=users,dc=my_co,dc=com

User template

cn=adminuser,cn=auxiliary,dc=my_co,dc=com

User search filter

Figure 5-137 Creating a realm

Entering information into the directory database

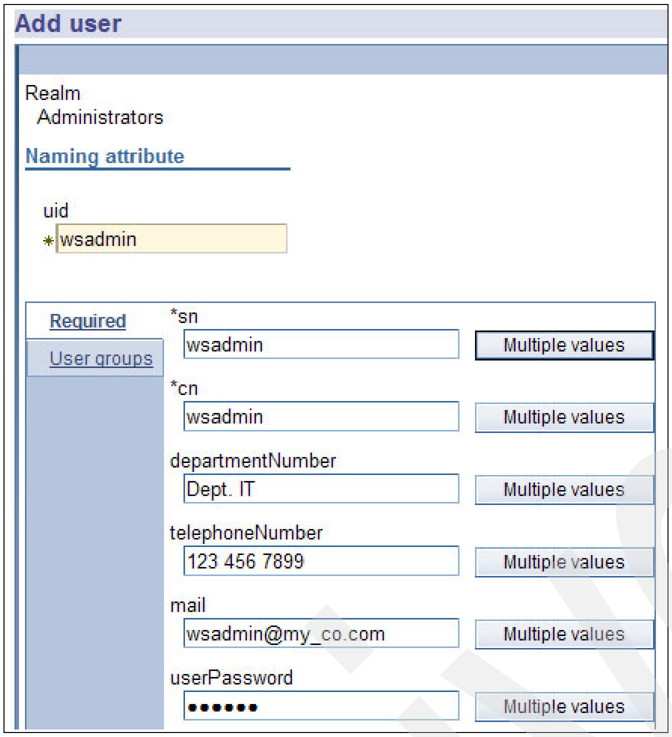
Now, you can add some users to your Directory server. We show an example of adding two users. Follow the next steps to incorporate the *wsadmin* and *wsbind* users to the directory:

1. In the Web Administration tool, click **Users and groups** → **Add user**.

2. From the Realm list, select **Administrators**. Click **Next**.

3. In the Add user panel (Figure 5-138), complete the following items:

   a. In the uid field, type `wsadmin`.
   b. In the *sn field type `wsadmin`.
   c. In the *cn field, type `wsadmin`.
   d. In the telephoneNumber field, type `123 456 7899`.
   e. In the departmentNumber field, type `Dept. IT`.
   f. In the mail field, type `wsadmin@my_co.com`.
   g. In the userPassword field, type `secret`.
   h. Click **Finish**.



Add user

Realm
Administrators

Naming attribute

uid
* wsadmin

*sn
wsadmin

*cn
wsadmin

departmentNumber
Dept. IT

telephoneNumber
123 456 7899

mail
wsadmin@my_co.com

userPassword
●●●●●●

Figure 5-138 Adding a new user

4. Repeat steps from 1 to 3, replacing *wsadmin* with `wsbind` to add wsbind user to the LDAP.
5. Log out of the Web administration tool by clicking **Logout** in the left navigation pane.

Testing the directory database

After you enter the administrator's data into the directory database, test the directory database and Directory Server.

1. Open a 5250 emulation to your Directory server system and signon.
2. On the command line, type `STRQSH` and press Enter.
3. Enter the following command to retrieve a list of all the LDAP entries in the Directory server under a certain subtree:

   ```
   ldapsearch –h myiSeries.my_co.com –b dc=my_co,dc=com objectclass=*
   ```

   In this command, note the following explanation:

   – *–h* is the name of the host machine running the LDAP server.
   – *–b* is the base DN to search under.
   – *objectclass=** returns all of the entries in the directory.

   Example 5-5 shows an example of the output from running this command.

Example 5-5 Sample output of the ldapsearch command

```
dc=my_co,dc=com
dc=my_co
objectclass=domain
objectclass=top

cn=groups,dc=my_co,dc=com
objectclass=container
objectclass=top
```

```
cn=groups

cn=auxiliary,dc=my_co,dc=com
objectclass=container
objectclass=top
cn=auxiliary

cn=admin,cn=groups,dc=my_co,dc=com
objectclass=groupOfNames
objectclass=top
member=cn=wsadmin
cn=admin

cn=adminUser,cn=auxiliary,dc=my_co,dc=com
javaSerializedData=NOT ASCII
objectClass=top
objectClass=javaContainer
objectClass=javaObject
objectClass=javaSerializedObject
javaClassNames=com.ibm.ldap.admin.usrAdmin.TemplateContainer
javaClassNames=java.lang.Object
javaClassNames=java.io.Serializable
javaClassName=com.ibm.ldap.admin.usrAdmin.TemplateContainer
cn=adminUser

cn=Administrators,cn=auxiliary,dc=my_co,dc=com
ibm-realmUserTemplate=cn=adminuser,cn=auxiliary,dc=my_co,dc=com
ibm-realmUserContainer=cn=users,dc=my_co,dc=com
objectclass=ibm-realm
objectclass=ibm-staticGroup
objectclass=top
ibm-realmGroupContainer=cn=groups,dc=my_co,dc=com
ibm-realmAdminGroup=cn=admin,cn=groups,dc=my_co,dc=com
ibm-realmUserSearchFilter=
cn=Administrators

uid=wsadmin,cn=users,dc=my_co,dc=com
departmentNumber=Dept. IT
mail=wsadmin@my_co.com
uid=wsadmin
objectclass=top
objectclass=inetOrgPerson
objectclass=organizationalPerson
objectclass=person
sn=wsadmin
telephoneNumber=123 456 7899
cn=wsadmin

uid=wsbind,cn=users,dc=my_co,dc=com
departmentNumber=Dept. IT
mail=wsbind@my_co.com
uid=wsbind
objectclass=top
objectclass=inetOrgPerson
objectclass=organizationalPerson
objectclass=person
sn=wsbind
telephoneNumber=123 456 7890
cn=wsbind
```

```
cn=users,dc=my_co,dc=com
objectclass=container
objectclass=top
cn=users
$
```

The first line of each entry is called the distinguished name. DNs are like the complete name of each entry. Some of the entries do not contain data and are only structural. Those with the line *objectclass=inetOrgPerson* correspond to the entries that you created for the WebSphere administrators.

5.7.3 Directory server replication overview

Replication provides two main benefits:

- Redundancy of information: Replicas back up the content of their supplier servers.
- Faster searches: Search requests can be spread among several different servers, all having the same content, instead of a single server. This improves the response time for the request completion.

Specific entries in the directory are identified as the roots of replicated subtrees, by adding the ibm-replicationContext objectClass to them. Each subtree is replicated independently. The subtree continues down through the directory information tree (DIT) until it reaches the leaf entries or other replicated subtrees. Entries are added below the root of the replicated subtree to contain the replication topology information. These entries are one or more replica group entries, under which are created replica subentries. Associated with each replica subentry are replication agreements. These agreements identify the servers that are supplied (replicated to) by each server and define the credentials and schedule information.

Through replication, a change made to one directory is propagated to one or more additional directories. In effect, a change to one directory shows up on multiple different directories. The IBM Directory supports an expanded master-subordinate replication model. Replication topologies are expanded to include:

- Replication of subtrees of the DIT to specific servers
- A multi-tier topology referred to as *cascading replication*
- Assignment of the server role (master or replica) by subtree
- Multiple master servers, referred to as *peer-to-peer replication*

The advantage of replicating by subtrees is that a replica does not need to replicate the entire directory. It can be a replica of a part, or subtree, of the directory.

The expanded model changes the concept of master and replica. These terms no longer apply to servers, but rather to the roles that a server has in regard to a particular replicated subtree. A server can act as a master for some subtrees and as a replica for others. The term, *master*, is used for a server that accepts client updates for a replicated subtree. The term, replica, is used for a server that only accepts updates from other servers designated as a supplier for the replicated subtree.

There are three types of directories as defined by function.

- Master/peer

  The master/peer server contains the master directory information from where updates are propagated to the replicas. All changes are made and occur on the master server, and the master is responsible for propagating these changes to the replicas. There can be several servers acting as masters for directory information, with each master responsible for updating other master servers and replica servers. This is referred to as *peer replication*. Peer replication can improve performance and reliability. Performance is improved by providing a local server to handle updates in a widely distributed network. Reliability is improved by providing a backup master server ready to take over immediately if the primary master fails.

> **Note:** Master servers replicate all client updates, but do not replicate updates received from other masters. Updates to the same entry made by multiple servers might cause inconsistencies in directory data because there is no conflict resolution.

- Cascading (forwarding)

  A *cascading server* is a replica server that replicates all changes sent to it. This contrasts with a master/peer server in that a master/peer server only replicates changes that are made by clients that are connected to that server. A cascading server can relieve the replication workload from the master servers in a network, which contains many widely dispersed replicas.

- Replica (read-only)

  This is an additional server that contains a copy of directory information. The replicas are copies of the master (or the subtree that it is a replica of). The replica provides a backup of the replicated subtree.

If the replication fails, it is repeated even if the master is restarted. You can use the *Replication Management* → *Manage queues* option for your Directory server in the Web administration tool to check for failing replication.

You can request updates on a replica server, but the update is forwarded to the master server by returning a referral to the client. If the update is successful, the master server then sends the update to the replicas. Until the master has completed replication of the update, the change is not reflected on the replica server where it was originally requested. Changes are replicated in the order in which they are made on the master.

If you are no longer using a replica, you must remove the replication agreement from the supplier. Leaving the definition causes the server to queue up all updates and use unnecessary directory space. Also, the supplier continues trying to contact the missing consumer to retry sending the data.

Building LDAP-enabled networks and applications is a common practice in enterprise applications. WebSphere is LDAP-enabled. When the LDAP server fails, WebSphere cannot access directory data, such as security data, and fails to service client requests. Therefore, building a HA LDAP is a part of the highly available WebSphere system.

Using clustering software and LDAP master-replica

iSeries LDAP supports a master and replica architecture that makes it possible for you to configure a HA LDAP without a switchable resource. Install clustering software on both nodes, and configure LDAP to use local data.

The primary node is configured as the LDAP master, and the backup node is configured as the LDAP replica, as shown in Figure 5-139. Any LDAP change requests that go to the replica server are referred to the master server because the replica server cannot change data. The master server sends all changes to the replica server to synchronize its data.

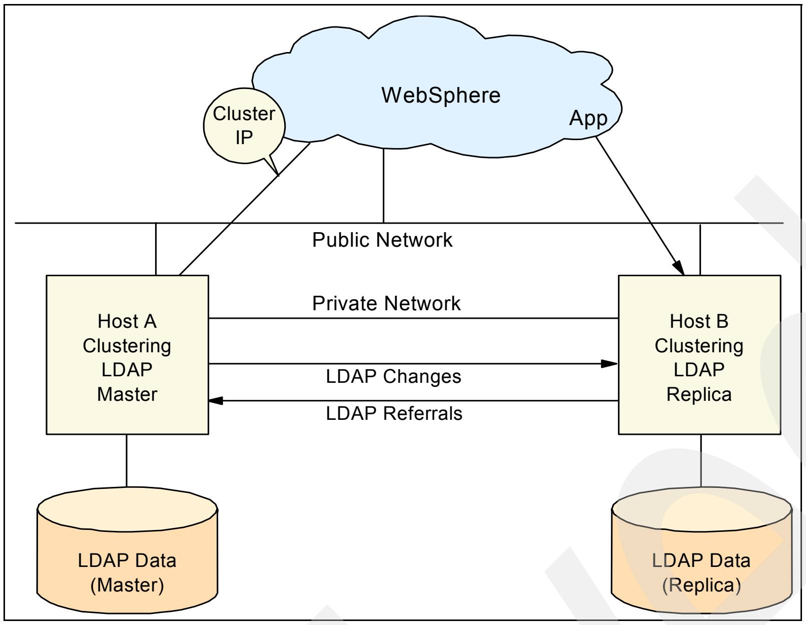


Figure 5-139 Clustered master-replica LDAP with individual disks

When the primary server (master) is down due to a network, hardware, or software reason, the LDAP service is moved to the backup server under the control of the clustering software (see Figure 5-140). By default, the replica server is *read-only*, so you can't modify the Directory Server data. While you cold reconfigure the replica server as a master, we recommend that you use the multi-master (also known as "peer-to-peer") configuration. See the following sections for more details.

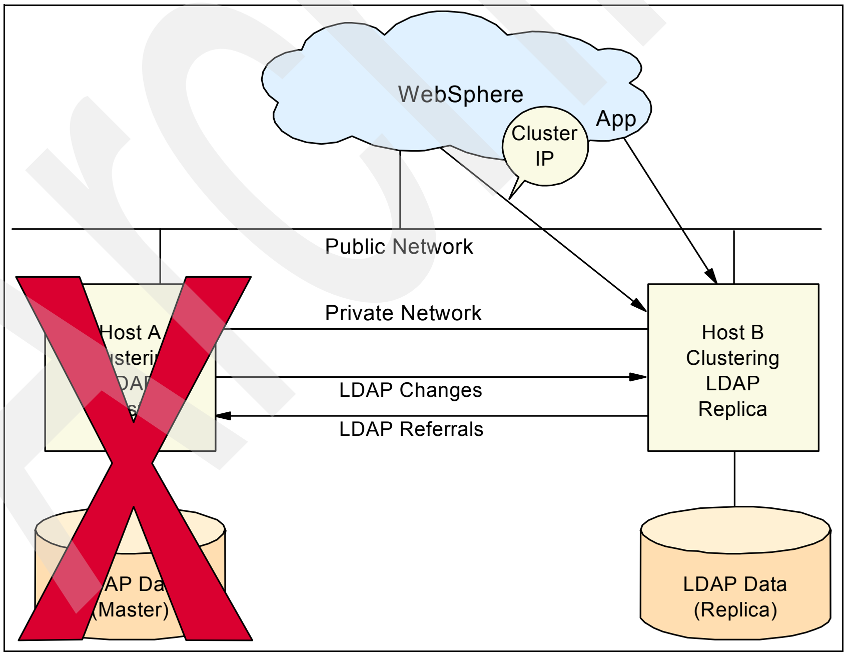


Figure 5-140 Clustered master-replica LDAP with individual disks after failover

When the primary node is up again, you can move the LDAP service back to the primary node.

You should not configure automatic fallback, because by doing so, you will lose all updates. You need to export the latest data from the backup server manually and import it to the primary server before you start the primary LDAP server again. It takes time to synchronize the data in the master server in this share-nothing configuration. In the shared disks LDAP configuration, because you use the same data in the shared disks, you do not need to synchronize the data between two servers. However, it is easier to configure the cluster without shared disks.

Using a network sprayer (Load Balancer)

In addition to configuring a HA LDAP with clustering software, you can build a low-cost, easy-to-configure HA LDAP with a network sprayer such as the WebSphere Edge Components' Load Balancer, or a Domain Name System (DNS) server that has a load balancing function (DNS round-robin), as shown in Figure 5-141.

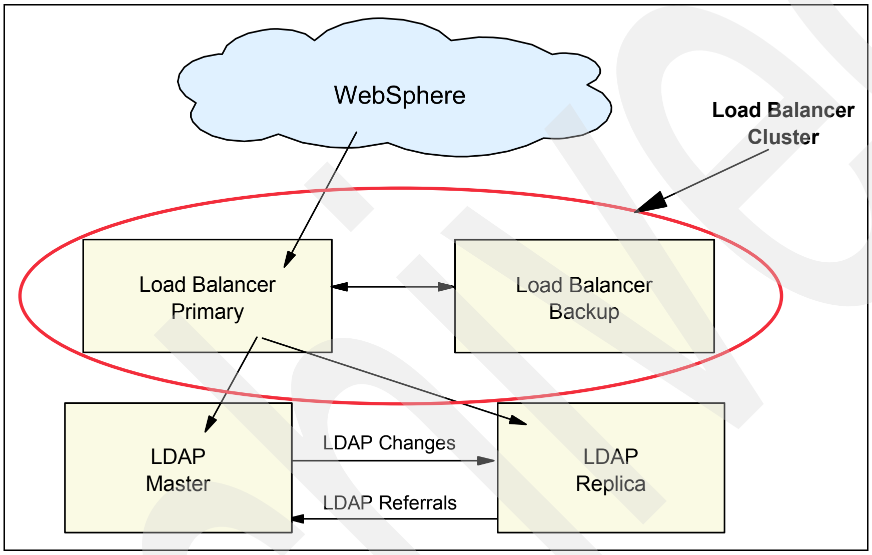


Figure 5-141 LDAP and Load Balancer, master-replica

The Load Balancer distributes client requests to both servers. When one LDAP server fails, the requests are directed to the other server, as shown in Figure 5-142.

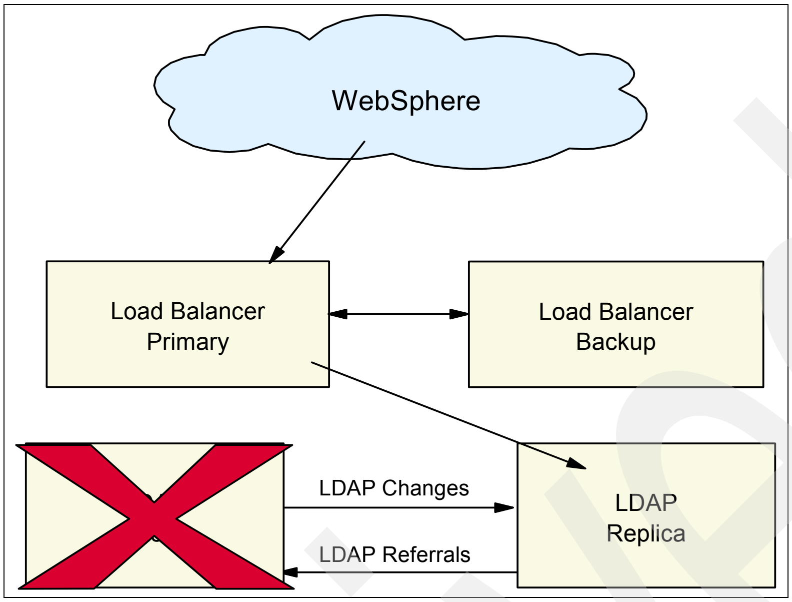


Figure 5-142 LDAP and Load Balancer after failure of an LDAP server

Because the Load Balancer has a backup configured, this system also caters to a failure of the Load Balancer as shown in Figure 5-143.

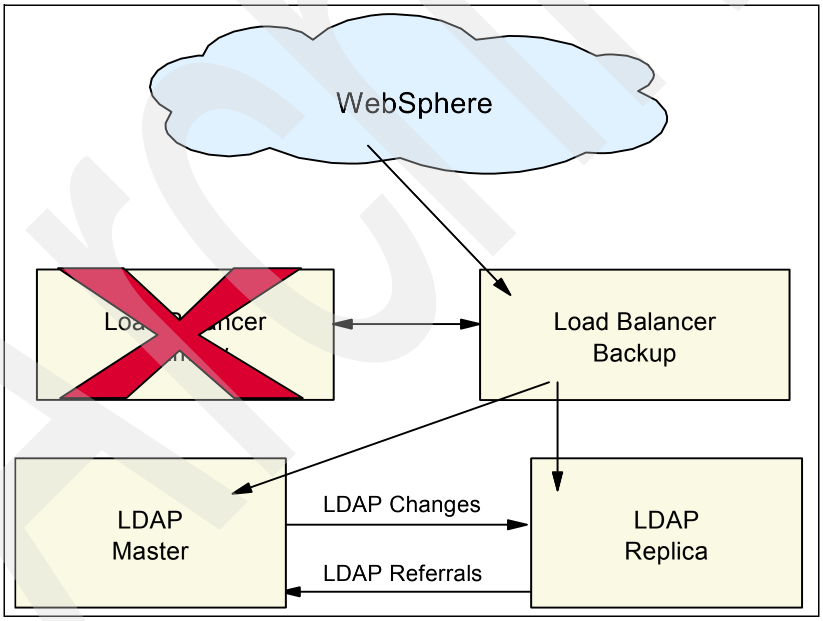


Figure 5-143 LDAP and Load Balancer after failure of the Load Balancer server

Using a network sprayer (Load Balancer) with LDAP peer replication (multi-master)

This setup is similar to the previous one with the exception that both LDAP servers are masters. It is possible to have several servers acting as masters for directory information, with each master responsible for updating other master servers and replica servers. This is referred to as *peer replication* (see Figure 5-144).

Peer replication can improve performance, availability, and reliability. Performance is improved by providing a local server to handle updates in a widely distributed network. Availability and reliability are improved by providing a backup master server ready to take over immediately if the primary master fails. Peer master servers replicate all client updates to the replicas and to the other peer masters, but do not replicate updates received from other master servers.

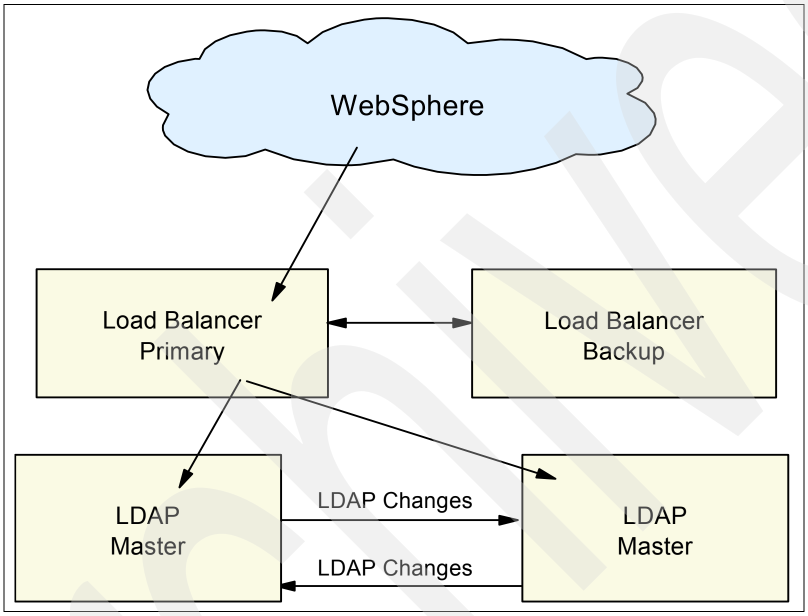


Figure 5-144 LDAP and Load Balancer, multi-master/peer replication

Note: If a high volume of directory changes occur in a brief interval of time, then consider using an “always true” Load Balancer rule and adding one LDAP server to that rule. This will ensure that all requests get directed to only *one* LDAP server while still ensuring that the other LDAP server can service requests in the event of a failure of the designated LDAP server associated with the always true rule.

Next step

The next section takes you step-by-step through creating a master/replica topology and a master/peer topology. It does not explain how to create a forwarding server. Different combinations of masters, replicas, and forwarding servers are possible.

For more details, look in the path **iSeries Information Center** → **eBusiness and Web serving** → **Security and Directory Server** → **Directory Server (LDAP)** → **Administer Server** and click **Manage replication** in the iSeries Information Center at:

http://publib.boulder.ibm.com/infocenter/iseries/v5r3/index.jsp

5.7.4 Creating a master-replica topology

To define a basic master-replica topology, you must follow these steps:

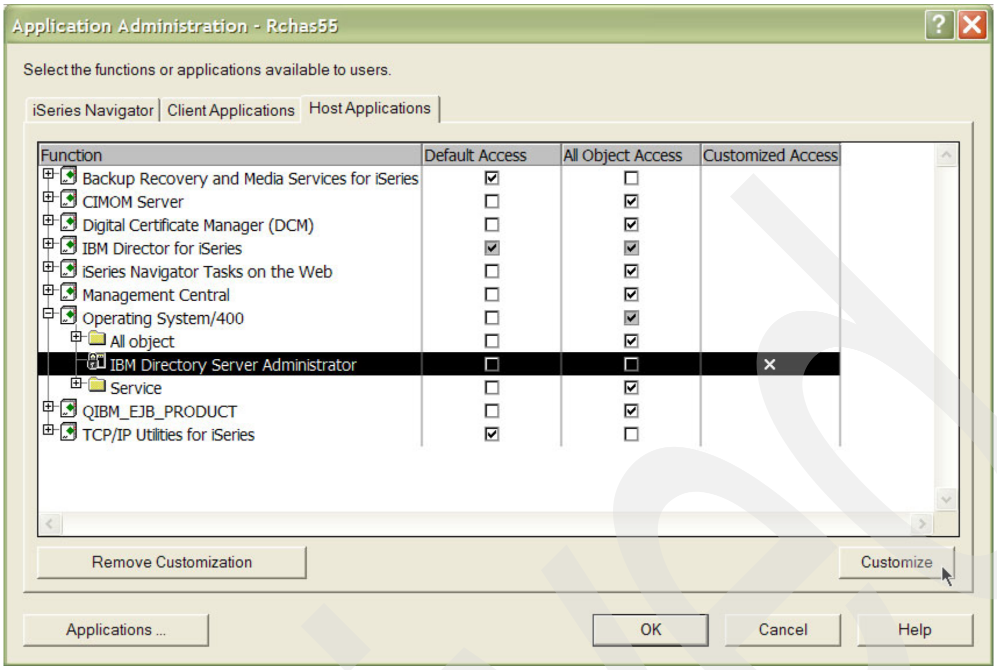
1. Create a master server and define what it contains. Select the subtree that you want to replicate and specify the server as the master.
2. Create credentials to be used by the supplier.
3. Create a replica server.
4. Export the topology from the master to the replica.
5. Change the replica's configuration to identify who is authorized to replicate changes to it, and add a referral to a master.

Before you create your topology, use the steps as explained in the following sections to properly set up the authorities for the user who will manage this topology.

Working with administrative access for authorized users

To avoid future authority problems when configuring LDAP topologies, follow these steps:

1. In iSeries Navigator, expand ***your system*** → **Network** → **Servers** and click **TCP/IP**. Right-click **IBM Directory Server** and select **Properties**.
2. On the General tab, under Administrator information, select the **Grant administrator access to authorized users** option. Click **OK**.
3. In the window that opens, select **Restart the server now**. Click **OK**.
4. Wait for the Directory Server to restart.
5. In iSeries Navigator, right-click ***your system name*** and select **Application Administration**.
6. If you are doing these steps for the first time, you are prompted to select the applications that your want to administer. Select all of them.
7. In the Application Administration window (Figure 5-145) that opens, complete these tasks:
   a. Click the **Host Applications** tab.
   b. Expand **Operating System/400**, and click **IBM Directory Server Administrator** to highlight the option.
   c. Click the **Customize** button.



Application Administration - Rchas55

Select the functions or applications available to users.

Function	Default Access	All Object Access	Customized Access
Backup Recovery and Media Services for iSeries	☑	☐	
CIMOM Server	☐	☑	
Digital Certificate Manager (DCM)	☐	☑	
IBM Director for iSeries	☑	☑	
iSeries Navigator Tasks on the Web	☐	☑	
Management Central	☐	☑	
Operating System/400	☐	☑	
All object	☐	☑	
IBM Directory Server Administrator	☐	☐	×
Service	☐	☑	
QIBM_EJB_PRODUCT	☐	☑	
TCP/IP Utilities for iSeries	☑	☐	

Figure 5-145 Selecting Directory server administrator function

d. In the window that opens, follow these steps:

i. Expand **All Users**, **Groups**, or **Users not in a Group tree**, whichever is appropriate for the user you want to give the administrative rights for the Directory server.

ii. Select a user or group to be added to the Access allowed list (must have *ALLOBJ and *IOSYSCFG special authorities).

iii. Click the **Add** button.

iv. Click **OK** to save the changes.

e. Back on the Application Administration window, click **OK**.

8. From now on, use the user that you added and its password (with the following convention) to login to the Web administration console and perform configuration tasks:

```
os400-profile=myuser,cn=accounts,os400-sys=systemA.mycompany.com
```

Here *myuser* is the user you selected in step 7 d ii, and *systemA.mycompany.com* is the fully qualified host name of your LDAP iSeries server. This user is known as a *projected user*.

Projected user: An LDAP entry representing an i5/OS user profile is referred to as a projected user. You can use the DN of a projected user along with the correct password for that user profile in a simple bind. For example, the DN for user JSMITH on system my-system.acme.com would be os400-profile=JSMITH,cn=accounts,os400-sys=my-system.acme.com.

The system objects' suffix in this example is os400-sys=my-system.acme.com. You can see it in the Properties of the IBM Directory Server, Database/Suffixes tab.

Creating a master server (replicated subtree)

> **Note:** The server must be running to perform this task.

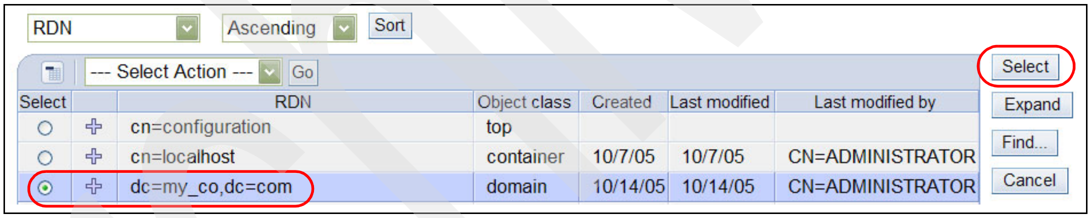
Creating a master server designates an entry as the root of an independently replicated subtree and creates an ibm-replicasubentry representing this server as the single master for the subtree. To create a replicated subtree, you must designate the subtree that you want the server to replicate. To perform this tasks, follow these steps:

1. In the Web Administration tool, expand **Replication management** and click **Manage topology**.
2. Click **Add subtree**.
3. The master server referral URL is displayed in the form of an LDAP URL, for example:

   ```
   ldap://myservername.mylocation.mycompany.com
   ```

   > **Note:** The master server referral URL is optional. It is used only:
   >
   > - If the server contains (or will contain) any read-only subtrees
   > - To define a referral URL that is returned for updates to any read-only subtree on the server

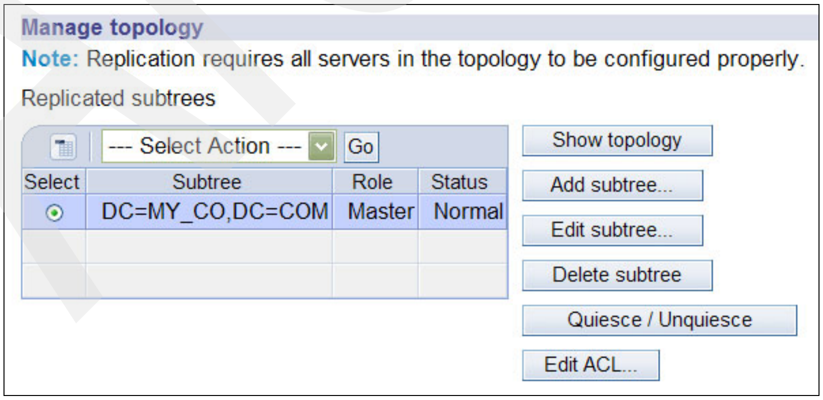
4. Enter the DN of the root entry of the subtree that you want to replicate or click **Browse** to select the entry that is to be the root of the subtree
5. As shown in Figure 5-146, select the replication subtree and click **Select**. Then click **OK**.



Select		RDN	Object class	Created	Last modified	Last modified by
○	✣	cn=configuration	top			
○	✣	cn=localhost	container	10/7/05	10/7/05	CN=ADMINISTRATOR
●	✣	dc=my_co,dc=com	domain	10/14/05	10/14/05	CN=ADMINISTRATOR

Figure 5-146 Selecting the replication subtree

6. The new server is displayed on the Manage topology panel, under the heading Replicated subtrees (see Figure 5-147). Click **Close**.


Manage topology

Note: Replication requires all servers in the topology to be configured properly.

Replicated subtrees

Select	Subtree	Role	Status
●	DC=MY_CO,DC=COM	Master	Normal

Figure 5-147 Added subtree

Creating credentials

Now you need to create the replication credentials. They are used to authenticate the master server to the replica server during the replication process.

1. Expand the **Replication management** category in the navigation area of the Web administration tool and click **Manage credentials**.
2. Select the location that you want to use to store the credentials from the list of subtrees. The Web administration tool allows you to define credentials in these locations:
   - cn=replication,cn=localhost, which keeps the credentials only on the current server

     > **Note:** In most replication cases, locating credentials in cn=replication,cn=localhost is preferred because it provides greater security than replicated credentials located on the subtree. However, there are certain situations in which credentials located on cn=replication,cn=localhost are not available.
     >
     > If you are trying to add a replica under a server, for example serverA, and you are connected to a different server with the Web administration tool, serverB, the Select credentials field does not display the option cn=replication,cn=localhost. This is because you cannot read the information or update any information under cn=localhost of the serverA when you are connected to serverB.
     >
     > The cn=replication,cn=localhost option is available only when the server under which you are trying to add a replica is the same server that you are connected to with the Web administration tool.

   - Within the replicated subtree, in which case the credentials are replicated with the rest of the subtree

     Credentials placed in the replicated subtree are created beneath the ibm-replicagroup=default entry for that subtree.

     > **Note:** If no subtrees are displayed, go to "Creating a master server (replicated subtree)" on page 215 to learn how to create the subtree that you want to replicate.

   Select **cn=replication,cn=localhost**, and click **Add**.
3. Enter the name for the credentials you are creating, for example, `cn=mycreds`.
4. Select the type of authentication method you want to use. Possible authentication methods are:
   - Simple bind authentication
   - Kerberos authentication
   - SSL with certificate authentication

   We chose **Simple bind**. Click **Next**.
5. If you selected the simple bind authentication, complete the following steps in the Add credential panel (Figure 5-148).

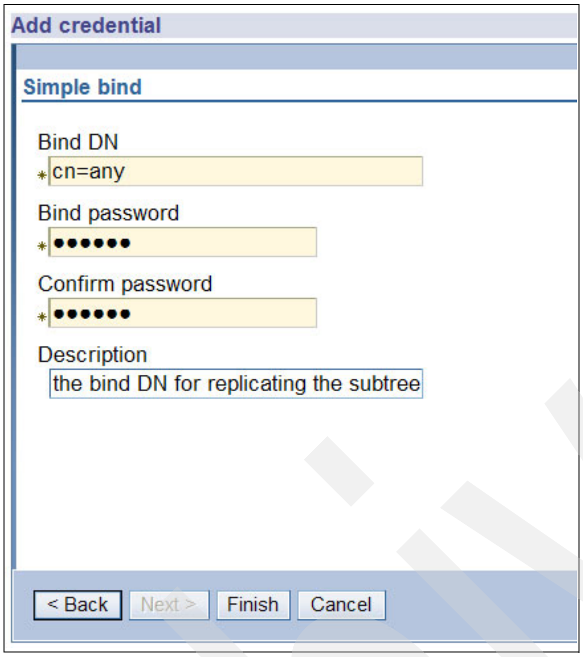
   a. For Bind DN, enter the DN that the server uses to bind to the replica, for example, `cn=any`.

   b. In the Bind password field, type the password that the server uses when it binds to the replica, for example, `secret`.

c. In the Confirm password field, type the password again to confirm that there are no typographical errors.

d. If you want, in the Description field, type a brief description of the credentials.

e. Click **Finish**.



Add credential

Simple bind

Bind DN: cn=any

Bind password: ●●●●●●

Confirm password: ●●●●●●

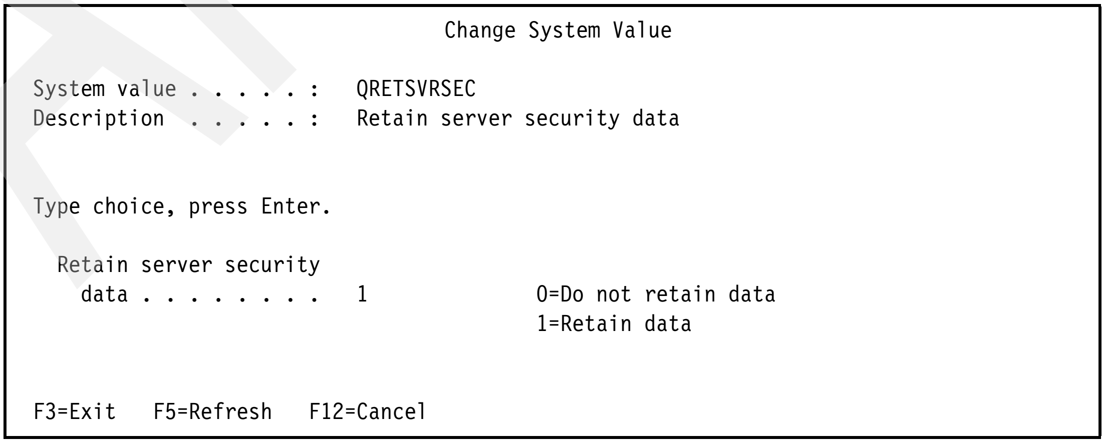
Description: the bind DN for replicating the subtree

Figure 5-148 Creating bind credentials

> **Note:** You might want to record the credential's bind DN and password for future reference. You need this password when you create the replica agreement.

6. Restart the IBM Directory Server to enable the changes.

7. On the server where you created the credentials, set the *Retain server security data* (QRETSVRSEC) system value to 1 (retain data) as shown in Figure 5-149. Since the replication credentials are stored in a validation list, this allows the server to retrieve the credentials from the validation list when it connects to the replica.



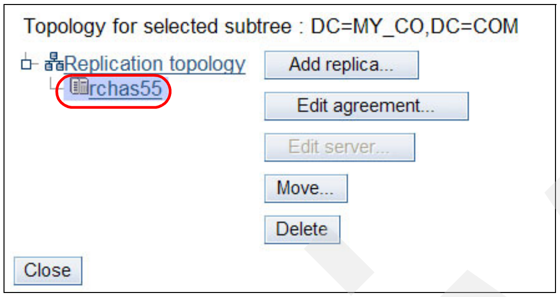
```
                                   Change System Value

System value . . . . . :   QRETSVRSEC
Description  . . . . . :   Retain server security data


Type choice, press Enter.

  Retain server security
    data . . . . . . . . .   1               0=Do not retain data
                                             1=Retain data


F3=Exit   F5=Refresh   F12=Cancel
```

Figure 5-149 Changing the QRETSVRSEC property

Creating a replica server

Note: The server must be running to perform this task.

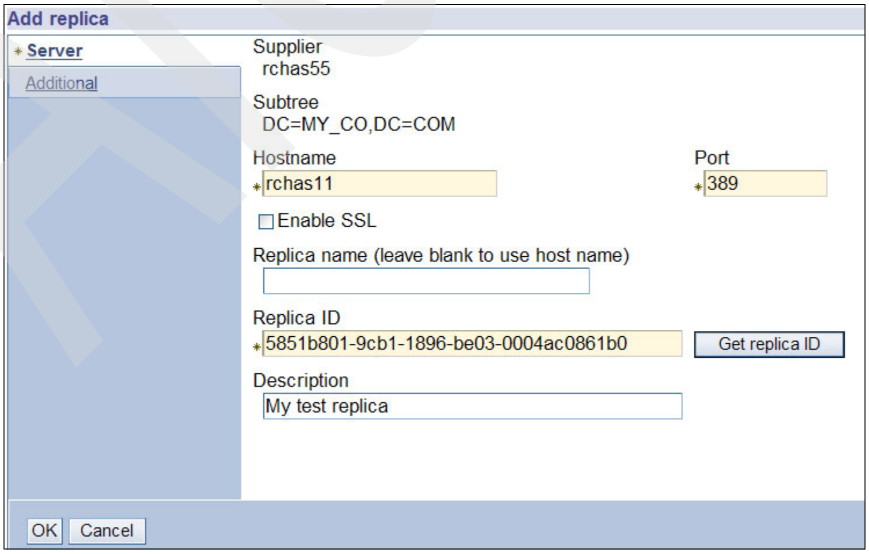
1. Expand the **Replication management** category in the navigation area and click **Manage topology**.
2. Select the subtree that you want to replicate and click **Show topology**.
3. Click the arrow next to the Replication topology selection to expand the list of supplier servers as shown in Figure 5-150. Click the supplier server, which is **rchas55** in our example. Click **Add replica**.



Topology for selected subtree : DC=MY_CO,DC=COM

Figure 5-150 Displaying the supplier server

4. The Add replica window opens. Click the **Server** link and complete the following steps.
   a. Enter the host name and port number for the replica you are creating. The default port is 389 for non-SSL and 636 for SSL. These are required fields.
   b. We don't enable SSL communications.
   c. Enter the replica name or leave this field blank to use the host name.
   d. Enter the replica ID. If the server on which you are creating the replica is running, click **Get replica ID** to automatically prefill this field. This is a required field, if the server you are adding is going to be a peer or forwarding server. We recommend that all servers be at the same OS/400 release.
   e. Enter a description of the replica server (see Figure 5-151).



Add replica

Supplier
rchas55

Subtree
DC=MY_CO,DC=COM

Hostname: rchas11

Port: 389

□ Enable SSL

Replica name (leave blank to use host name):

Replica ID: 5851b801-9cb1-1896-be03-0004ac0861b0

Description: My test replica

Figure 5-151 Defining a replica

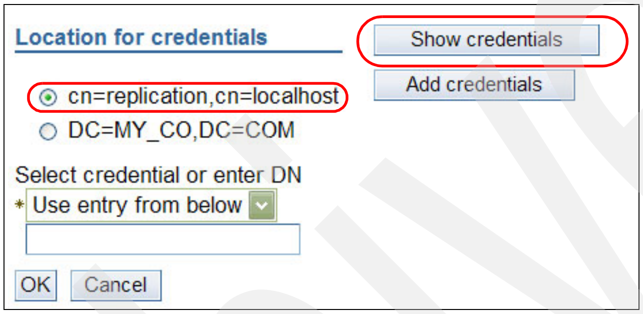
Click the **Additional** link (top left corner in the Add replica window). Specify the credentials that the replica uses to communicate with the master. Placing credentials in cn=replication,cn=localhost is considered more secure.

> **Note:** The Web administration tool allows you to define credentials in these places as explained in “Creating credentials” on page 216:
>
> - cn=replication,cn=localhost
> - Within the replicated subtree

a. Click **Select**.

b. Select the location for the credentials you want to use. Preferably this is **cn=replication,cn=localhost**.

c. Click **Show credentials** (see Figure 5-152).


Location for credentials

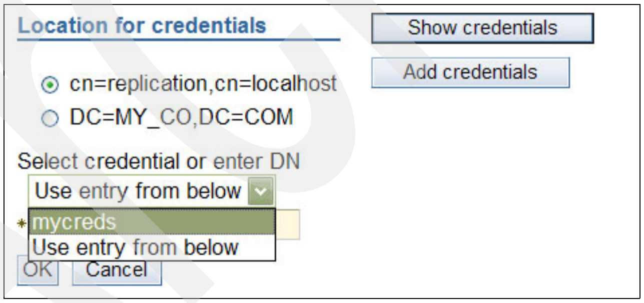
● cn=replication,cn=localhost
○ DC=MY_CO,DC=COM

Select credential or enter DN
* Use entry from below

Figure 5-152 Navigating to the credentials

d. Expand the list of credentials and select the one you want to use (see Figure 5-153).

e. Click **OK**. See “Creating credentials” on page 216 for additional information about agreement credentials.


Location for credentials

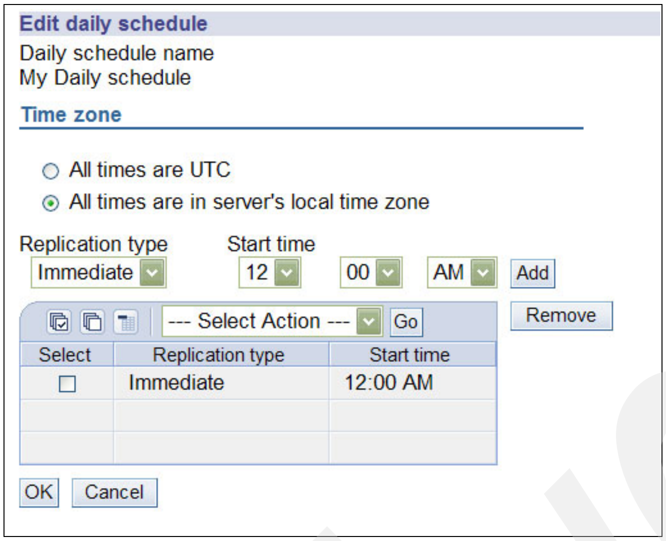
● cn=replication,cn=localhost
○ DC=MY_CO,DC=COM

Select credential or enter DN
Use entry from below
mycreds
Use entry from below

Figure 5-153 Selecting the credentials

5. Specify a replication schedule. In our example, we don’t have any schedule configured, so we need to create one. Click **Add**.
6. In the Weekly schedule name box type the name, `My schedule`, for example. Click **Add daily schedule**.
7. In the next display, type the name of the daily schedule.
8. Specify the replication time and click **Add** (see Figure 5-154). Using this example, the replication starts coping any outstanding updates prior to 12 AM and then continues in the real-time replication for the rest of the day. If we use this daily schedule for each day of the

week (as shown in Figure 5-155), we effectively create the real-time replication 24x7. This is a likely choice for HA solutions. Click **OK**.



Edit daily schedule

Daily schedule name
My Daily schedule

Time zone

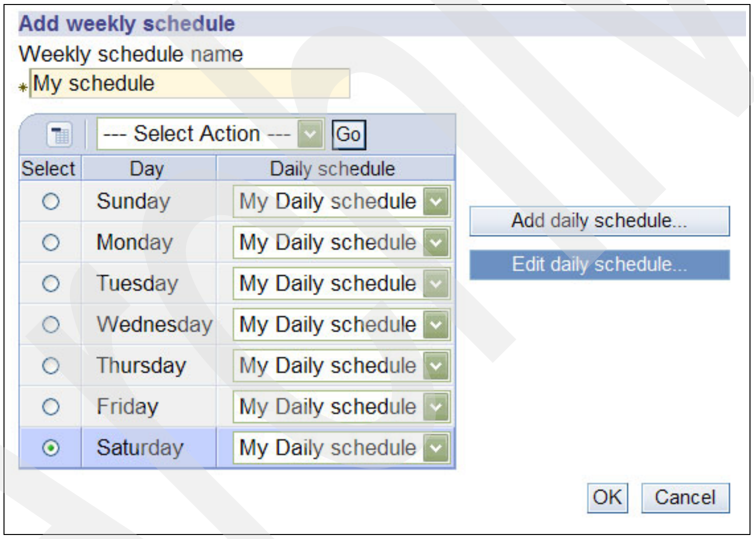
○ All times are UTC
● All times are in server's local time zone

Replication type: Immediate
Start time: 12 : 00 AM

Select	Replication type	Start time
□	Immediate	12:00 AM

Figure 5-154 Creating a daily schedule

9. In the Add weekly schedule panel (Figure 5-155), click **OK**.



Add weekly schedule

Weekly schedule name
* My schedule

Select	Day	Daily schedule
○	Sunday	My Daily schedule
○	Monday	My Daily schedule
○	Tuesday	My Daily schedule
○	Wednesday	My Daily schedule
○	Thursday	My Daily schedule
○	Friday	My Daily schedule
●	Saturday	My Daily schedule

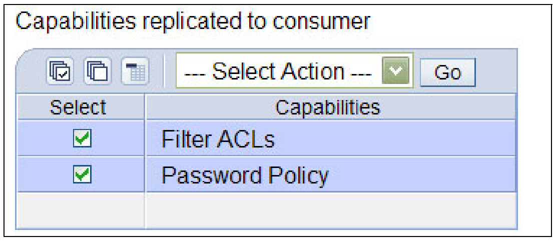
Figure 5-155 Weekly schedule display

10. Now you're back to the Additional tab of the replica configuration.

From the list of supplier capabilities, you can deselect any capabilities that you do not want replicated to the consumer (see Figure 5-156). If your network has a mix of servers at different releases, capabilities are available on later releases that are not available on earlier releases.

Some capabilities, such as filter ACLs and password policy, use operational attributes that are replicated with other changes. In most cases, if these features are used, you want all servers to support them. If all of the servers do not support the capability, you do not want to use it. For example, you might not want different ACLs in effect on each server.

However, there might be cases where you might want to use a capability on the servers that support it. And you might not have changes related to the capability replicated to servers that do not support the capability. In such cases, you can use the capabilities list to mark certain capabilities to not be replicated. Click **OK** to create the replica.



Capabilities replicated to consumer

Select	Capabilities
☑	Filter ACLs
☑	Password Policy

Figure 5-156 Replication capabilities

11.A message is displayed noting that additional actions must be taken. Click **OK**.

> **Note:** If you are adding more servers as additional replicas or are creating a complex topology, do not proceed with “Copying data to the replica” on page 221 or “Adding supplier information to the replica” on page 223 until you have finished defining the topology on the master server. If you create the masterfile.ldif after you complete the topology, the file contains the directory entries of the master server and a complete copy of the topology agreements. When you load this file on each of the servers, then each server has the same information.

Copying data to the replica

After you create the replica, you must export the topology from the master to the replica.

1. On the master server, create an LDIF file for the data. To copy all the data contained on the master server, complete these steps:

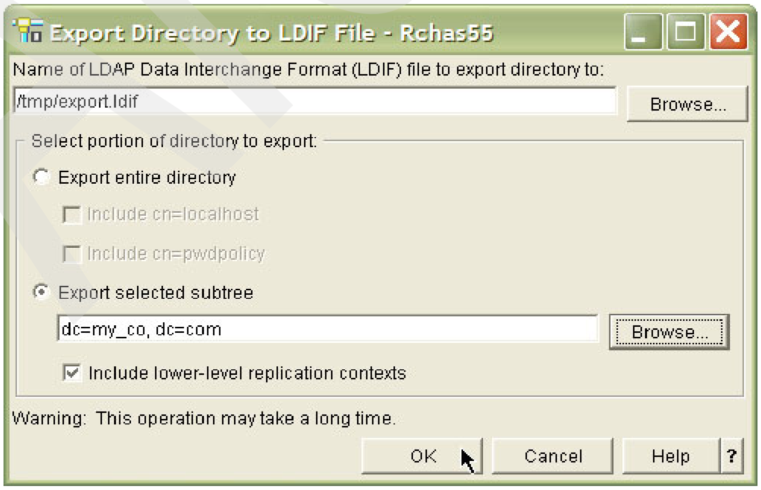
   a. In iSeries Navigator, expand ***your iSeries system*** → **Network** → **Servers**, and select **TCP/IP**. Right-click **IBM Directory Server** and select **Tools** → **Export File**.

   b. The Export Directory to LDIF File window (Figure 5-157) opens.

      i. Specify the output LDIF file name, for example `export.ldif`.

      ii. Select **Export a selected subtree** and specify a subtree to export. In our case, we type `dc=my_co,dc=com`.

      iii. Click **OK**.



Export Directory to LDIF File - Rchas55

Name of LDAP Data Interchange Format (LDIF) file to export directory to: /tmp/export.ldif

Select portion of directory to export:

○ Export entire directory

□ Include cn=localhost

□ Include cn=pwdpolicy

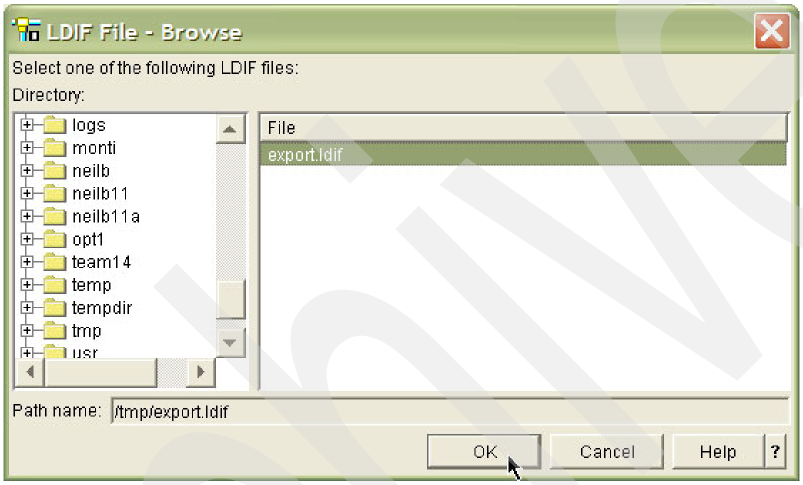
● Export selected subtree: dc=my_co, dc=com

☑ Include lower-level replication contexts

Warning: This operation may take a long time.

Figure 5-157 Exporting a subtree

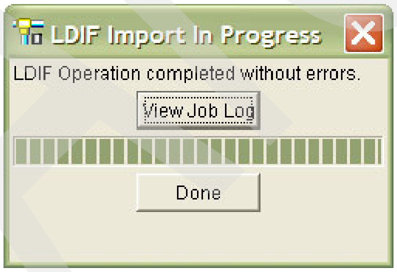
2. Move your LDIF file to the replica server. You can use iSeries Navigator drag-and-drop feature or FTP.
3. On the machine where you are creating the replica, complete these steps:
   a. Ensure that the replicated suffixes are defined in the replica server's configuration (see step 7 on page 198).
   b. In iSeries Navigator:
      i. Expand ***your iSeries system*** → **Network** → **Servers** and select **TCP/IP**.
      i. Right-click **IBM Directory Server** in the right pane and select **Stop**.
      ii. When the server stops, right-click **IBM Directory Server** again and select **Tools** → **Import File**.
   c. In the LDIF File - Browse window (Figure 5-158), navigate to the location of the LDIF file and click **OK**.



Path name: /tmp/export.ldif

Figure 5-158 Navigating to the LDIF file

   d. Click **OK** again.
   e. The LDIF Import in Progress window (Figure 5-159) opens. After the import has successfully completed, you should see that there are no errors. Click **Done**.



LDIF Operation completed without errors.

Figure 5-159 Status window

   f. Start the IBM Directory Server.

Adding supplier information to the replica

You need to change the replica's configuration to identify who is authorized to replicate changes to it, and add a referral to a master. On the machine where you are creating the replica, complete these steps:

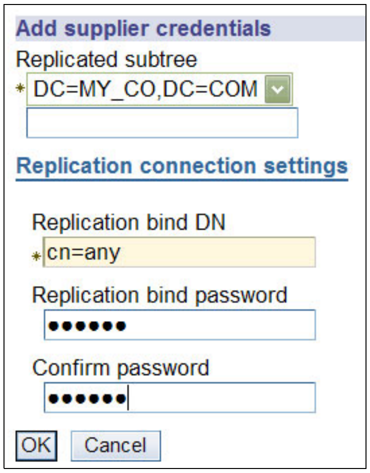
1. Open the Web Administration tool's login page.
2. Login to the Console Admin and add the iSeries system where you host the replica to the managed console servers (see step 4 on page 199 and step 7 on page 199).
3. Logout from the Console Admin.
4. Go back to the login page of the Web Administration tool and select your replica's host name from the LDAP Hostname drop-down list.
5. Enter the user ID and password for the Directory Server on this iSeries and click **Login**.

> **Important:** You must login as a *projected user*. You can use the DN of a projected user along with the correct password for that user profile in a simple bind. For example, the DN for user JSMITH on system my-system.acme.com would be:
>
> ```
> os400-profile=JSMITH,cn=accounts,os400-sys=my-system.acme.com
> ```
>
> The highlighted parts are specific to your environment.

6. Expand **Replication management** in the navigation area and click **Manage replication properties**.
7. Click **Add**.
8. Under Add supplier credentials, complete these tasks:
   a. Select a supplier from the Replicated subtree menu or enter the name of the replicated subtree for which you want to configure supplier credentials. If you are editing supplier credentials, this field is not editable. In our example, this field reads **DC=MY_CO,DC=COM**.
   b. Enter the replication bind DN. In this example, it's `cn=any`.

   > **Note:** You can use either of these two options, depending on your situation.
   >
   > - Set the replication bind DN (and password) and a default referral for all subtrees replicated to a server using *default credentials and referral*. You might use this option when all subtrees are replicated from the same supplier.
   > - Set the replication bind DN and password independently for each replicated subtree by adding supplier information for each subtree. You might use this option when each subtree has a different supplier (a different master server for each subtree).

   c. Depending on the type of credential, enter and confirm the credential password (recorded from step 5 on page 216). Click **OK**.



Add supplier credentials

Replicated subtree

* DC=MY_CO,DC=COM

Replication connection settings

Replication bind DN

* cn=any

Replication bind password

●●●●●●

Confirm password

●●●●●●

Figure 5-160 Providing credentials

9. Restart the replica's IBM Directory Server for the changes to take effect.

The replica is in a suspended state and no replication is occurring. After you finish setting up your replication topology, verify your replication as explained in "Verifying the replication queues" on page 227. The replica now receives updates from the master.

5.7.5 Creating a master-peer topology

Peer replication is a replication topology in which multiple servers are masters. However, unlike a multi-master environment, no conflict resolution is done among peer servers. LDAP servers accept the updates provided by peer servers and update their own copies of the data. No consideration is given for the order in which the updates are received, or whether multiple updates conflict.

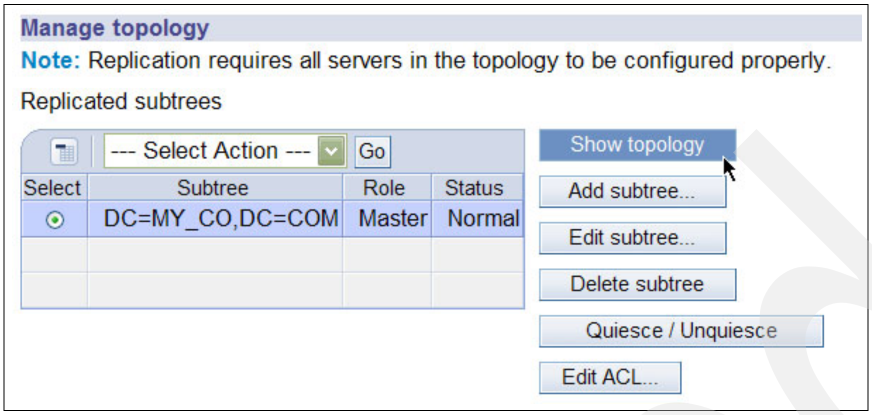
Use peer replication only in environments where the pattern of directory updates is well known. Updates to particular objects within the directory must be done *only* by one peer server. This is intended to prevent the scenario of one server deleting an object, followed by another server modifying the object. This scenario creates the possibility of a peer server receiving a delete command followed by a modify command, which creates a conflict.

Later, when you add a Load Balancer nodes for the failover support for the LDAP servers (see 5.7.7, "Adding Load Balancer to the LDAP servers topology" on page 228), you have to configure the Load Balancer to use one LDAP master server at a time (a *primary* server). Only when the primary LDAP server fails, the Load Balancer switches over to the second LDAP server (a *peer server*). With this configuration, you resolve the potential problem of making two updates for the same object at the same time.

To add more masters (peers), you add the server as a read-only replica of the existing masters, initialize the directory data, and then promote the server to be a master.

1. Make sure that you have completed all the steps as provided in 5.7.4, "Creating a master-replica topology" on page 213.
2. When this topology works, you need to promote a replica server to become a second master (a *peer server*). Access the Web Administration tool's login page.
3. Select your master server in the LDAP Hostname list, provide your credentials (as a projected user, see the Note box on page 223), and click **Login**.
4. Expand **Replication management** and click **Manage topology**.

5. Select your replication subtree and click **Show topology** (see Figure 5-161).



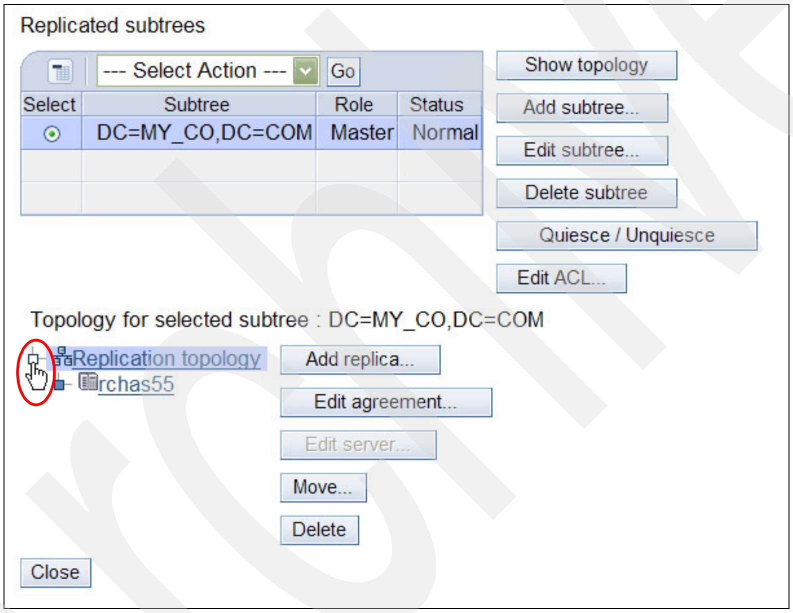
Note: Replication requires all servers in the topology to be configured properly.

Replicated subtrees

Select	Subtree	Role	Status
●	DC=MY_CO,DC=COM	Master	Normal

Figure 5-161 Displaying a topology

6. Expand **Replication topology** by clicking the small square next to it (see Figure 5-162).



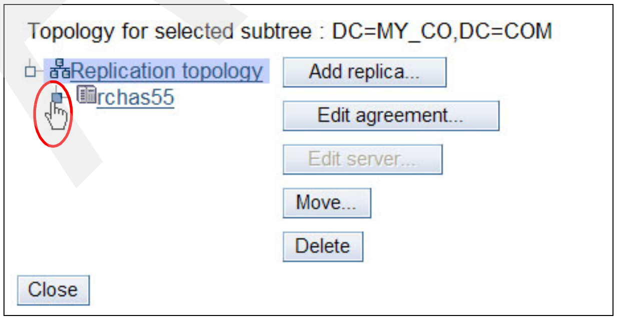
Replicated subtrees

Select	Subtree	Role	Status
●	DC=MY_CO,DC=COM	Master	Normal

Topology for selected subtree : DC=MY_CO,DC=COM

Figure 5-162 Expanding the topology

7. Expand your master node (see Figure 5-163).



Topology for selected subtree : DC=MY_CO,DC=COM

Figure 5-163 Expanding the master node

8. Select your replica node and click **Move** (see Figure 5-164).

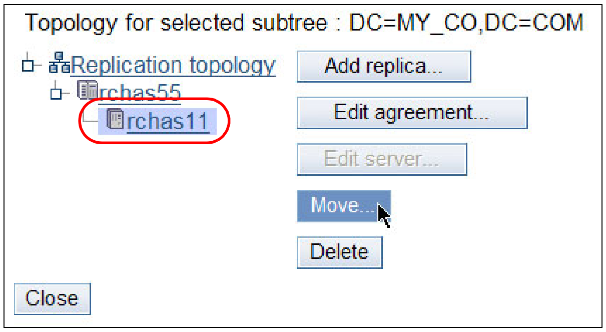
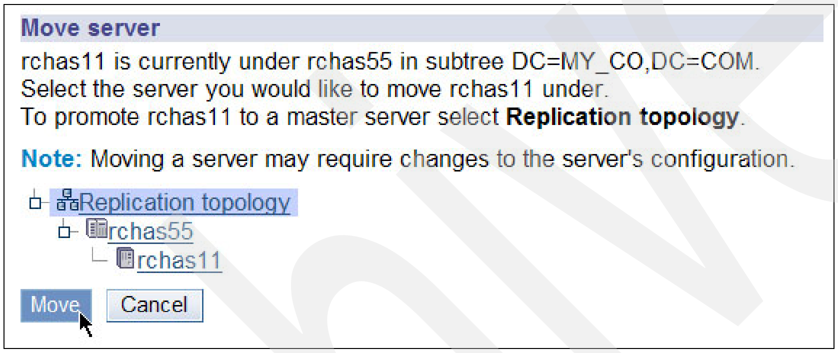


Figure 5-164 Selecting the replica node

9. In the next panel (Figure 5-165), make sure that **Replication topology** is selected and click **Move**.



Move server

rchas11 is currently under rchas55 in subtree DC=MY_CO,DC=COM.
Select the server you would like to move rchas11 under.
To promote rchas11 to a master server select **Replication topology**.

Note: Moving a server may require changes to the server's configuration.

Figure 5-165 Promoting a replica to the master

10. In the next panel, the tool allows your to remove any unwanted agreements between newly created *peer server* and the original master server. In our example, we accept the default agreement and click **Continue**.

> **Note:** Select the supplier agreements that are appropriate for the role of the server. For example, if a replica server is being promoted to be a peer server, you must select to create supplier agreements with all the other servers and their first-level replicas. These agreements enable the promoted server to act as a supplier to the other servers and their replicas. Existing supplier agreements from the other servers to the newly promoted server are still in effect and do not need to be recreated.

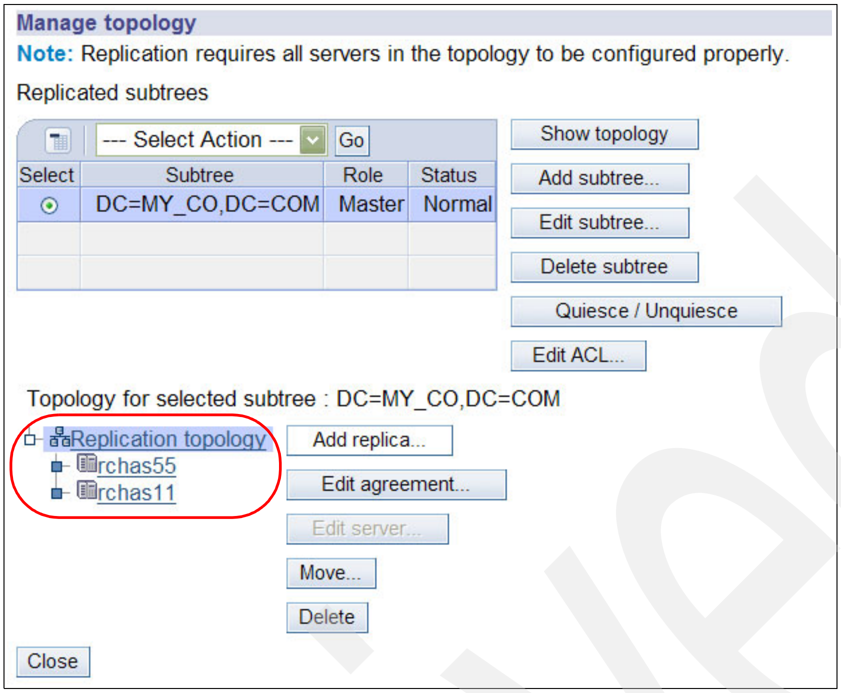
11. In the next panel, the tool informs you that it will collect information for agreements from the other master server. Click **OK**.

12. In the Select credentials panel, click **Add credentials**.

13. Create credentials that will be used for the replication from the new peer server to the original master server. Type the name of the credentials, for example `cn=new creds`. Use **Simple bind** as the authentication method. Click **Next**.

14. Provide the credential values. In our example, we use the same values as for the original master server, `cn=any` with password of `secret`. Click **Finish**.

15. In the next panel, In the Select credentials or enter DN box, select the newly created credentials and click **OK**. You should now see that the topology view has changed, so that both servers are peers (see Figure 5-166).



Manage topology

Note: Replication requires all servers in the topology to be configured properly.

Replicated subtrees

Select	Subtree	Role	Status
●	DC=MY_CO,DC=COM	Master	Normal

Topology for selected subtree : DC=MY_CO,DC=COM

Figure 5-166 Topology view changed

16.Add the same credentials to replicate from the new peer server to the original master server. See "Adding supplier information to the replica" on page 223. You must perform this step on the original master server (rchas55 in our example) using a *projected user ID*.

17.Restart IBM Directory Server on both iSeries.

Now, if you look at the Manage queues panel, on both LDAP servers in the Web Administration tool, you see the *Ready* status (see Figure 5-167).

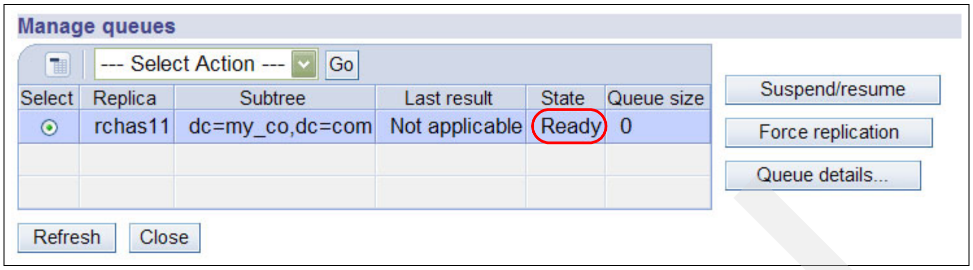
Verifying the replication queues

Verify the replication queues to be sure that the replication between the servers is working properly. Follow these steps from the console of both peers (or master server) to check that the replication is working.

1. Using the Web Administration tool, log in to the LDAP server.
2. Expand **Replication management**, and select **Manage queues**.
3. The state of the replication queue should be *Ready*.
   - If it is not, click **Suspend/resume** (see Figure 5-167).
   - If it does not change to Ready, search for messages in the QDIRSRV job log.
     i. In the 5250 window, enter the following command:

     ```
     WRKJOB JOB(QDIRSRV)
     ```

     ii. Select the job in the active status (1=Select).
     iii. Type 10 (Option: Display job log, if active or on job queue) and press Enter.
     iv. Continue as indicated in the messages text.
4. Log out from the console.



Select	Replica	Subtree	Last result	State	Queue size
●	rchas11	dc=my_co,dc=com	Not applicable	Ready	0

Figure 5-167 Verifying the replication status

5.7.6 Removing details of the previous configuration

If you are in a testing phase and have created different topologies and templates, realms, users, and groups, you might want to get rid of any trace of a previous configuration. However it is not enough to delete these configuration objects from the Directory Server console. In this case, you can follow the next procedure to delete it all and start from scratch:

1. End the server by typing the following command on the OS/400 command line:

   ```
   ENDTCPSVR SERVER(*DIRSRV)
   ```

2. Create a library.

   ```
   CLRLIB LIB(QUSRDIRDB)
   ```

   QUSRDIRDB is the default library for IBM Directory Server on iSeries. It stores the LDAP configuration information. You can verify it in iSeries Navigator by displaying the LDAP Server Properties.

3. Start the server.

   ```
   STRTCPSVR SERVER(*DIRSRV)
   ```

You now have a clean LDAP server configuration and can start from the beginning.

After you create your master-peer topology, you must add a failover support by using Load Balancer. See 5.7.7, “Adding Load Balancer to the LDAP servers topology” for more information.

5.7.7 Adding Load Balancer to the LDAP servers topology

OS/400 provides a capability to have a replica of the Directory Server content. However, it doesn’t support a switchover capability. To bypass this limitation, you need to use Load Balancer to distribute the LDAP workload. When you configure this support, keep in mind these important points (see Figure 5-3 on page 98):

- Several Load Balancer servers can be active in your overall network, provided that each one uses a different cluster domain address. For example, a typical Load Balancer that sprays to a Web server cluster uses a cluster domain address, such as www.MyCompany.com, that is accessible to Internet users. A second Load Balancer that sprays requests from WebSphere clients to a LDAP server cluster is likely located in a protected company network with the LDAP servers, so the cluster domain address is naturally different for that Load Balancer.
- In a mutual high-availability configuration for the IBM Load Balancer, in which two Load Balancers both actively handle requests and provide backup for each other, each one must use a different cluster domain address. For instance, one may use www1.MyCompany.com, and the other may use www2.MyCompany.com.

- If a large volume of directory server updates is expected to occur for Application Client requests that will be routed to a Load Balancer address, then configure the Load Balancer to spray only to one LDAP server while maintaining failover capability to another LDAP server in a multi-master directory server configuration. You can do this by defining an always-true rule in the load balancer configuration and then assigning one LDAP server to the rule. If the server assigned to the rule becomes unavailable, then the Load Balancer sprays requests to the other LDAP server in the cluster.
- When configuring a built-in or custom advisor for the Load Balancer for either port 80 or port 389 requests, we recommend that you set the connect timeout and receive timeout values to at least 60 seconds to prevent the Load Balancer from marking a server down during peak usage intervals.
- The clients that send requests to a Load Balancer cluster address must not be located on any of the clustered servers defined in the Load Balancer configuration. Those client requests then bypass Load Balancer. This is especially important to remember when WebSphere is the client. WebSphere and an LDAP server should not be collocated if a Load Balancer is supposed to forward LDAP requests sent from a WebSphere application.
- If you use Load Balancer for load distribution in front of an IBM Directory Server, Tivoli® Access Manager policy server, or Tivoli Access Manager authorization server, we recommend that you increase stale time-out for that port. By default, stale time-out is set to 300 seconds and should be set in minimum to 3600 (recommendation within the PMR). For LDAP, the recommendation is 28800 seconds, as indicated in the documentation for Load Balancer. For Tivoli Access Manager, there is no documented recommendation.
- After an IBM Directory Server takeover, it takes up to 120 seconds before you can access LDAP again. During this time, the WebSphere Application Server cell hangs. We found that this duration is related to the LDAP search time-out which can be specified in the LDAP settings of WebSphere Application Server. To reduce the duration of the hang, you can decrease the LDAP search time-out, but do not make it too small or you will not get anything back from LDAP. Or you can disable the connection option in the LDAP settings of WebSphere Application Server.
- If you want to spray the load between different LDAP servers by using Load Balancer, you have to disable the connection option in the LDAP settings of WebSphere Application Server.

5.8 Functional verification testing

To test the complex scenario, you must verify all components. This includes the edge servers, HTTP servers, application server, and IASP switchover. The verification test is conducted based on the Trade6 application, which is installed on the cluster. Trade6 uses a database, which allows the testing of the IASP switchover.

To test the environment in a similar real-life scenario, you need stress tools. Stress tools allow for the simulation of an appropriate load. A number of stress tools are available such as Rational Performance Tester, OpenSTA tool, and Apache JMeter.

Next you must prepare the stress access plan. There are a few parameters to be aware of such as the number of concurrent users, the delay between subsequent hits, and warm-up time. In our stress test, we used the following parameters and corresponding values:

- Hit address: `http://edgeserver/trade/scenario`
- Number of concurrent users: `25`
- Delay between hits: `3 seconds`
- Delay time deviation: `1 second`

The Delay time deviation parameter allows the addition of a random amount of time, like in a real-life scenario.

You must also be concerned about the warm-up time, which indicates the time allowed for the environment to stabilize. Stabilizing needs to occur for the database connection pool to be created, JSP pages to be compiled, and caches to be filled of the application servers.

To be sure that all the data gathered during the stress test is correct, you need to warm up the application. The amount of time for an application to warm up depends on the environment and the application, but in general, it is safe to assume that a few minutes will be needed. The importance of the warm-up process is shown in Figure 5-168. The blue line shows an average response time. The response time decreases until some saturation state. After this saturation point is reached, the response time stabilizes. The most important point to consider is that all performance tests should be run after the environment is stabilized.

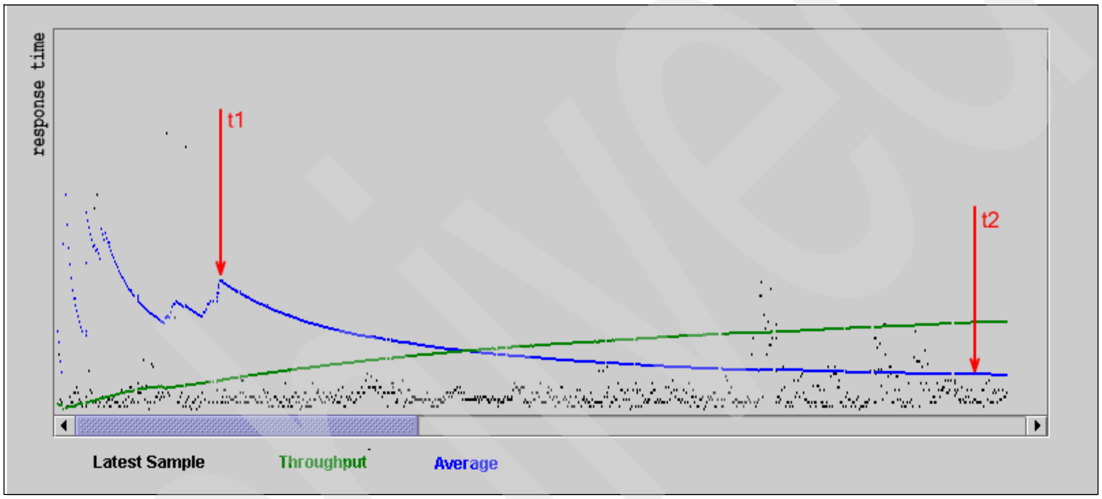


Figure 5-168 Environment warm up

The first component in the topology is the Edge Server. Although there are a few components in the Edge Server, the test focuses only on Load Balancer. The main feature of the load balance is to spread requests from Internet users among all available Web servers. In this scenario, two Web servers, located on iSeries servers, are used.

Figure 5-169 shows a typical load distribution on the Web servers. It shows that Load Balancer spreads request between the two Web servers. An average load on both Web servers is typically the same.

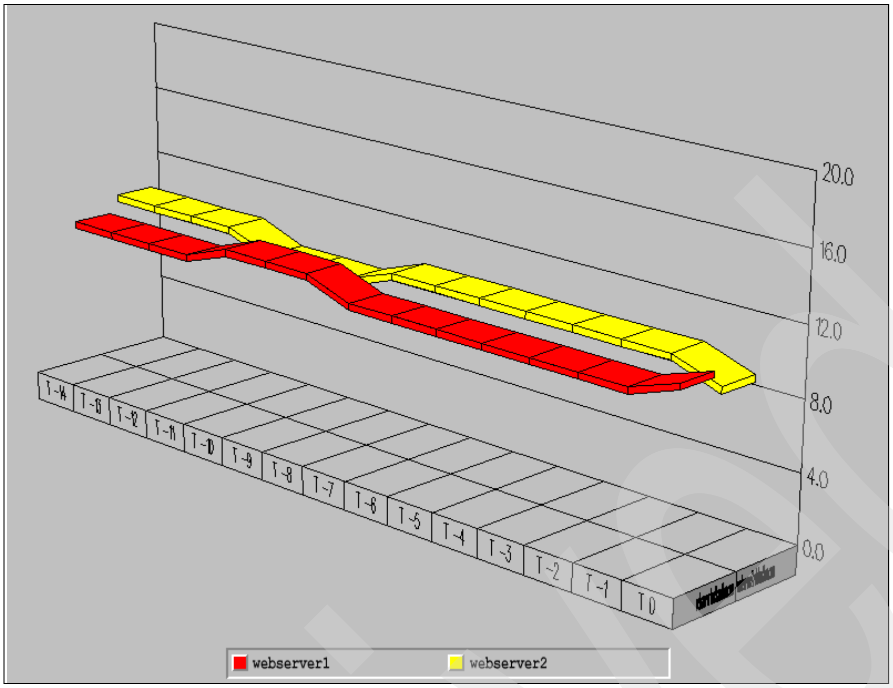


Figure 5-169 Load Balancer balancing requests between Web servers

Another Load Balancer responsibility is to check the availability of the Web servers. Figure 5-170 shows a particular case in which one of the Web servers is not available. All the Internet requests are then routed to the WEBSERVER2 iSeries which, in this case is the only Web server available. Figure 5-170 also shows that when the first Web server became unavailable, the second Web server acquired additional load.

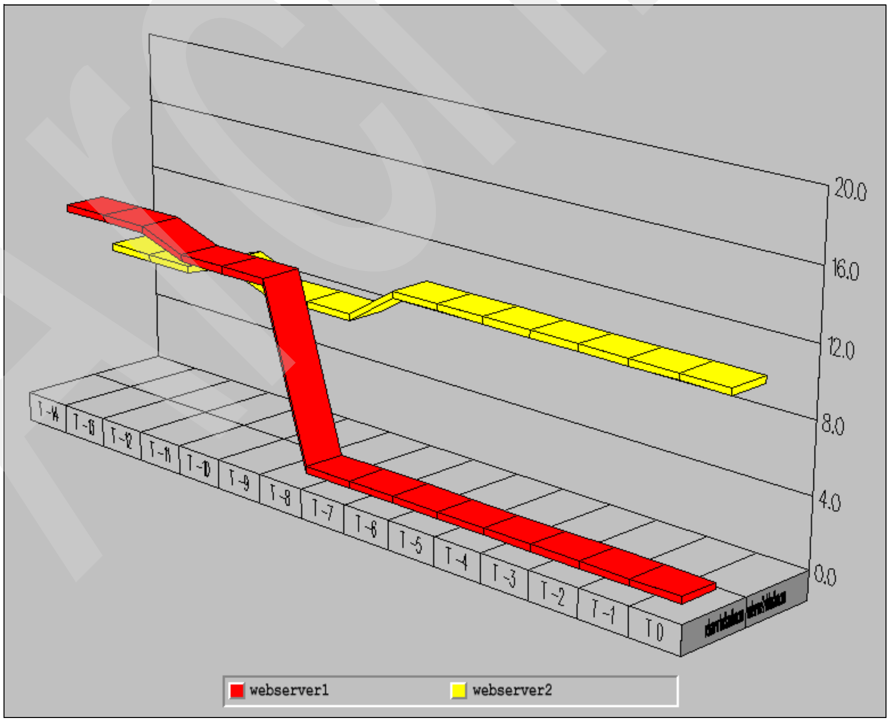


Figure 5-170 Load Balancer checking availability

Figure 5-171 shows the response time with two of the four cluster members down. Two areas in Figure 5-171 are represented by p1 and p2. These areas show a stable state where all four cluster member are operational. A period of time, shown between the t1 and t2 time marks, represent an unstable area. This area is a time slice during which the cluster members are shutting down. In this period, some requests may be lost. When the Web server's plug-in identifies these two cluster members as unavailable, they are excluded from the requests list. The blue curve represents the average response time. Figure 5-171 shows that, because of the two unavailable cluster members, the average response time is increased.

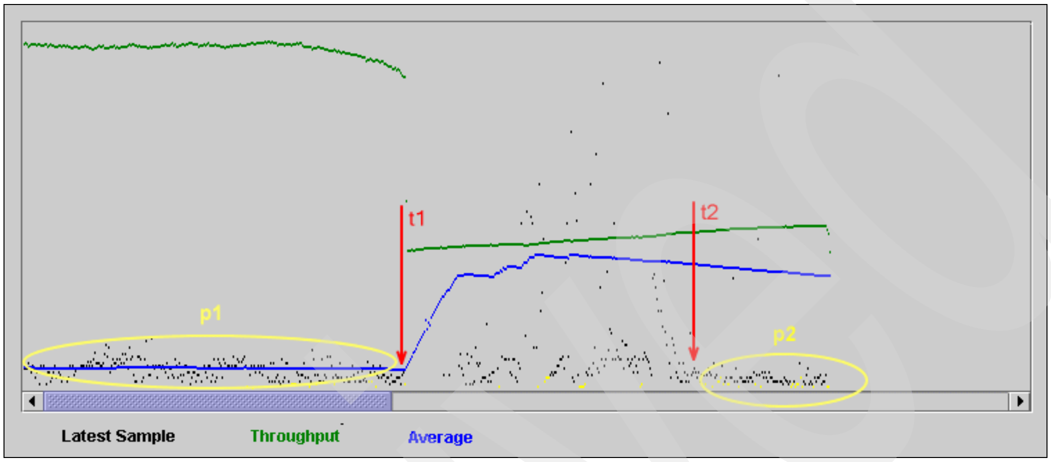


Figure 5-171   Response time with two cluster members down

One final test that you need to conduct is an XSM test. In our scenario, we force the switching of a database from one IASP to the other. This process, which is forced by the operator, is called *switchover*. During switchover, the database is not available for a period of time. From a user perspective, the time during which the application is unavailable doesn't depend on the database replication speed, but on the time-out values defined within this configuration. Figure 5-172 depicts the response time to the application before, during, and after switchover. The scale on the Y-axis is logarithmic because of the large difference in response time during normal operational state of the environment and the response times during switchover.

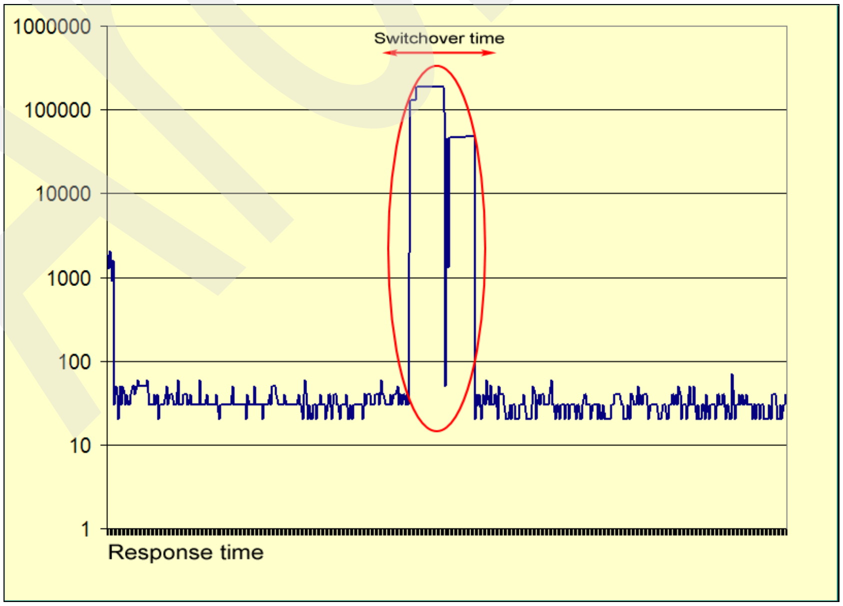


Figure 5-172   Switchover response times

Part 3

Appendixes

This part includes information that does not directly instruct you on how to build a WebSphere topology. However this information is essential for successful adoption of the concepts presented this book, for your environment.

A

WebSphere Application Server: New capabilities in Version 6

This appendix presents the new capabilities of WebSphere Application Server V6. The information in this chapter was compiled using materials from the following publications:

- *WebSphere Application Server V6: Planning and Design WebSphere Handbook*, SG24-6446
- WebSphere Application Server for OS/400 V6.0 Information Center

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

Programming model

This section addresses the enhancements to the programming model for WebSphere Application Server V6.

Application programming model support

WebSphere Application Server V6 supports the new Java 2 Platform, Enterprise Edition (J2EE), 1.4 specification. It also supports the previous versions 1.3 and 1.2 programming models. Migrating existing J2EE 1.2 or 1.3 applications to run in WebSphere Application Server V6 is greatly simplified.

Similar to WebSphere Application Server V5, you must have a V4 data source for your J2EE 1.2 application that needs access to a back-end relational database.

Programming model extensions

Programming model extensions (PMEs) are IBM-developed extensions to the J2EE model. In version 6 of WebSphere, PMEs that were formerly available only in the Enterprise Edition of WebSphere are now included in all versions of WebSphere 6. The core extensions include:

- Last Participant Support
- Internationalization Service
- WorkArea Service
- ActivitySession Service
- Extended JTA Support
- Startup Beans
- Asynchronous Beans
- Scheduler Service (Timer Service)
- Object Pools
- Dynamic Query
- Application Profiling
- DistributedMap
- Programming Model (with migration support)
- Web Services Gateway Filter

Changes in Web services

WebSphere Application Server V6 contains a Web services engine, which supports the J2EE 1.4 standards for applications. The Web services engine was designed to both support JSR 101 (known as *JAX-RPC*) and JSR 109 (known as *Enterprise Web services*).

To increase performance, the Web services engine in now based on Simple API for XML (SAX) instead of the previous Apache AXIS technology. To help with development, Web services are tightly integrated into the IBM Rational Applicational Developer tool and application server support. WebSphere V6 also provides support to integrate Web services with JMS to ensure message delivery.

Web services support has been updated to include the latest in technology options.

- *Java API for XML-based RPC (JAX-RPC) 1.1* enables you to develop Simple Object Access Protocol (SOAP)-based interoperable and portable Web services and Web service clients. The JAX-RPC programming model is defined by Web services standard JSR 101.
- *Web services for Java 2 Platform, Enterprise Edition* (JSR 109 - WSEE) defines the programming model and run-time architecture for implementing Web services based on the Java language.

- *SOAP with Attachments API for Java (SAAJ) 1.2* is used for SOAP messaging that works behind the scenes in the JAX-RPC implementation.
- *Web Services Security (WS-Security)* proposes a standard set of SOAP extensions that you can use to build secure Web services. Web Services-Interoperability (WS-I) Basic Profile 1.1 is a set of non-proprietary Web services specifications that promote interoperability.
- *Web Services-Interoperability (WS-I) Attachments Profile* compliments the WS-I Basic Profile 1.1 to add support for interoperable SOAP messages with attachments-based Web services.
- *Java API for XML Registries (JAXR) 1.0* defines a Java client API for accessing both Universal Description, Discovery, and Integration (UDDI, Version 2 only) and ebXML registries.
- *UDDI V3* is a private UDDI registry that implements V3.0 of the UDDI specification. This registry enables the enterprise to run its own Web services broker within the company or to provide brokering services to the outside world. The UDDI registry installation and management is now integrated with WebSphere Application Server.

In addition, WebSphere Application Server V6 adds value to the standards in these ways:

- Custom bindings to supplement JAX-RPC features, allowing you to create your own custom bindings to map Java to XML and XML to Java conversions
- Support for generic SOAP elements
- Multiprotocol support for a stateless session Enterprise JavaBean (EJB) as the Web service provider for enhanced performance without changes to JAX-RPC clients

Service Data Object

Service Data Object (SDO), formerly *WebSphere Data Objects*, provides unified data access and representation across heterogeneous data stores. With SDO, data mediators perform the real work of accessing the data stores. Clients query a data mediator service and get a data graph in response. The data graph consists of structured data objects representing the data store. Clients update the data graph and send it back to the mediator service to have the updates applied. SDO is not intended to replace other data access technologies, but rather to provide an alternate choice. It has the advantage of simplifying the application programming tasks required to access data stores. SDO support is included in WebSphere Studio Application Developer 5.1.1 and in Rational Application Developer 6.0. This support includes:

- Wizards and views for working with data objects
- Relational data lists
- Relational data objects

WebSphere Application Server 6.0 support for SDO includes:

- Support for SDO naming and packaging
- Externalization of the SDO Core API
- EJB Mediator for entity EJBs

JDBC Data Mediator for relational databases supported by WebSphere Application Server SDO is defined by JSR 235. For more information, see:

`ftp://www6.software.ibm.com/software/developer/library/j-commonj-sdowmt/Commonj-SDO-Specification-v1.0.doc`

JavaServer Faces

JavaServer Faces (JSF) is a user interface framework or API that eases the development of Java-based Web applications. JSF makes J2EE more approachable to non-Java application developers with HTML, scripting, and page layout skills. WebSphere Application Server version 6.0 supports JavaServer Faces 1.0 at a runtime level. There is also a support for the JSF development in WebSphere Studio Application Developer 5.1.1 and in Rational Application Developer 6.0.

System management

WebSphere Application Server V6 has enhanced the usability of the administration tools and introduced additional features for managing multiple instances of WebSphere. It has also addressed changes to application deployment in this version. WebSphere Application Server V6 provides improvements on the flexible and open System Management model, from WebSphere Application Server V5, with many new enhancements.

Mixed cell support

Mixed cell support allows you to run mixed versions of the application server within a V6 Network Deployment Manager cell. This is a convenient feature for migrating existing V5 Network Deployment cells to a V6 environment. When migrating, this feature allows you to first migrate Deployment Manager from V5 to V6 and continue to run your existing V5 application servers in the new V6 cell until you migrate them to V6. Application servers from V5 and V6 can coexist in a V6 Network Deployment cell. Deployment Manager V6 runs in compatibility mode by default, where it can manage both V5 nodes and V6 nodes.

Support for J2EE 1.4 specification

The J2EE 1.4 specification added several requirements for application server vendors in support of administration. This revision of the J2EE specification adds the following requirements to support:

- Java Management Extensions (JMX™) 1.2
- J2EE Management Specification (JSR-077)
- J2EE Connector Architecture (JCA) 1.5

In addition to J2EE specification-related administration features, the embedded messaging component of WebSphere Application Server that supports Java Messaging Service (JMS) has been redesigned to be significantly better integrated with the application server administration.

Java Management Extensions (JMX 1.2)

JMX is a framework that provides a standard way of exposing Java resources, like application servers, to a system management infrastructure. Using the JMX framework, a provider can implement such functions as listing the configuration settings and editing the settings. This framework also includes a notification layer that management applications can use to monitor events such as the startup of an application server.

The key features of the WebSphere Application Server V6 implementation of JMX include:

- All processes that run the JMX agent
- All run-time administration that is performed through JMX operations

- Connectors that are used to connect a JMX agent to a remote JMX-enabled management application

  The following connectors are supported:

  - SOAP JMX Connector
  - Remote Method Invocation over the Internet Inter-ORB Protocol (RMI-IIOP) JMX Connector
- Protocol adapters that provide a management view of the JMX agent through a given protocol

  Management applications that connect to a protocol adapter are usually specific to a given protocol.
- The ability to query and update the configuration settings of a run-time object
- The ability to load, initialize, change, and monitor application components and resources during run-time

Each Java Virtual Machine (JVM) in WebSphere Application Server includes an embedded implementation of JMX. In WebSphere Application Server V5, the JVMs contained an implementation of the JMX 1.0 specification. In WebSphere Application Server V6, the JVMs contain an implementation of the JMX 1.2 specification. The JMX 1.0 implementation used in WebSphere Application Server V5 is the TMX4J package that IBM Tivoli products supplied.

The JMX 1.2 specification used in WebSphere Application Server V6 is the open source MX4J package. The JMX implementation change across the releases does not affect the behavior of the JMX MBeans in the Application Server. No Application Server administrative APIs are altered due to the change from the JMX V1.0 specification to the JMX V1.2 specification.

The JMX V1.2 specification is backward compatible with the JMX 1.0 specification. However, you might need to migrate custom MBeans that are supplied by products other than WebSphere Application Server. The primary concern for these custom MBeans is related to the values that are used in key properties of the JMX ObjectName class for the MBean. The open source MX4J implementation more stringently enforces property validation according to the JMX 1.2 specification. Test the custom MBeans that you deployed in WebSphere Application Server V5 in V6 to ensure compatibility. You can find full details about the JMX V1.2 specification changes from the JMX V1.0 specification in the JMX 1.2 specification at:

http://java.sun.com/products/JavaManagement

J2EE management specification (JSR-077)

The management layer of the JMX architecture uses an implementation of the distributed services specification (JSR-077), which is not yet part of the J2EE specification. The management layer defines how external management applications can interact with the underlying layers in terms of protocols, APIs, and so on.

This specification enables third-party vendors to build the tools that manage the WebSphere configuration (similar to WebSphere Administrative Console).

J2EE Connector Architecture (JCA 1.5)

WebSphere Application Server V6 supports the JCA Version 1.5 specification, which provides new features such as the inbound resource adapter.

New administrative commands

A new wsadmin scripting object, *AdminTask*, is introduced in this release. Various AdminTask commands are implemented for important administrative scenarios such as server management, cluster management, and resource management. AdminTask commands provide various user friendly and task-oriented **wsadmin** commands. AdminTask commands may have multiple steps for some complicated administrative tasks similar to the wizard in the console. AdminTask commands are grouped based on their function areas.

You can find detailed help information for the AdminTask commands and command groups in the various AdminTask help commands. All AdminTask commands can be run in interactive mode, which leads users step by step, interactively. Then the corresponding AdminTask command is generated to help you learn the AdminTask command syntax.

WebSphere profiles

The concept of a WebSphere profile was introduced in WebSphere Application Server for iSeries long ago. However, in previous releases, it was referred to as a *WebSphere instance*. Each WebSphere profile uses separate configuration files but shares the binaries from which the profile was created. Each profile is distinguished by its base path, its own directory structure, and its own setupCmdLine script to configure its command-line environment.

A given configuration can be propagated from one WebSphere profile to another profile by exporting that profile to a configuration archive and importing it to another WebSphere profile (see “WebSphere configuration archive” on page 244). This mechanism works between WebSphere profiles of the same or different installations. A restriction is that this mechanism only works for unfederated WebSphere profiles.

Server templates

Server management functionality is enhanced significantly in this release, with the introduction of the server template. Clients can create a customized server template based on an existing server configuration and use them to create new servers. This provides clients a mechanism to propagate the server configuration within the same cell easily. Furthermore, clients can propagate the server configuration across the cell boundary by exporting a server configuration to a configuration archive then importing the archive to another cell.

Resource providers

For the following J2EE resources, used by applications, all functions are available for all scopes, including WebSphere Application Server V5.x and V6 nodes.

- Java Database Connectivity (JDBC) providers
- Generic JMS providers
- WebSphere embedded JMS providers
- WebSphere MQ JMS providers
- Mail providers
- Resource environment providers
- URL providers

The generic JMS provider is available for cell scope and WebSphere Application Server V6 nodes for backwards compatibility. For WebSphere Application Server V6 nodes, you are encouraged to use the WebSphere default messaging provider.

The following J2EE resources have limitations on WebSphere Application Server V5.0.x nodes:

- Resource adapters

  The format of a resource adapter configuration has been changed considerably to accommodate JCA 1.5 in J2EE 1.4. Both JCA 1.5 and JCA 1.0 resource adapters may be defined at the cell scope. However, JCA 1.5 adapters will not be available on a WebSphere Application Server V5.x node. For WebSphere Application Server V6 nodes and servers, both JCA 1.5 and JCA 1.0 resources may be defined. For WebSphere Application Server V5.x nodes and servers, only JCA 1.0 resource adapters may be defined.

- WebSphere default message provider

  The WebSphere default message provider is new in WebSphere Application Server V6. Its definitions may be created at the cell scope containing WebSphere Application Server V6 and V5.x nodes, but it will not be available on the WebSphere Application Server V5.x nodes. Its definitions can be created on any WebSphere Application Server V6 nodes or servers, but not on a WebSphere Application Server V5.x node or server.

Support for extensible server types

New in WebSphere Application Server V6 is the concept of adding Web server definitions to an application server definition in order to simplify associations with Web servers and application servers.

Web server

New in WebSphere Application Server V6 is the ability to define a Web server definition within an application server. In the case of the IBM HTTP server, the Web server can be managed from the WebSphere administrative tools. Updating of the plug-in configuration files is handled transparently for remote servers where the HTTP server resides. With the Web server being defined in the administrative process, applications can now be associated with one or more Web servers. This allows only specific Web servers the ability to serve selected applications. Web servers can be defined on managed or unmanaged nodes.

Managed nodes contain a node agent which allows the administrative process to have access to manage the Web server node. Management features include starting and stopping the HTTP server and automatic plug-in configuration file placement on the Web server node. Unmanaged nodes are defined in the administrative process, but do not have a node agent present.

Special consideration must be taken if there is a firewall between a Deployment Manager node and a managed node. Additional ports will have to be opened in the firewall to support the node agent process. For this reason, most managed nodes are placed behind a domain firewall. Unmanaged nodes are more likely to be placed in a demilitarized zone (DMZ) configuration.

Generic server

Defining a generic server in an application server profile allows you to associate this profile with a non-WebSphere server or process that is needed to support the application server environment. Examples of generic servers are a Java server, C or C++ server or process, CORBA server, or RMI server.

Introduction of node groups

New in WebSphere Application Server V6 is a feature called *node group*. A node group is a collection of managed nodes. It defines a boundary for server cluster information. Nodes that you organize into a node group should be enough alike in terms of installed software, available resources, and configuration to enable servers, on those nodes, to host the same applications as part of a server cluster.

The primary reason of adding a node group is to support z/OS and other platforms as part of the same WebSphere cell.

Improved administrative console look and feel

The WebSphere Application Server V6 Administrative Console appearance and functionality has been improved. The console views have changed based on the context being displayed. The console has been updated in V6 to comply and resemble the look and feel of other cross-IBM products.

New panels have been added to facilitate the new V6 features such as service integration, the integrated UDDI Registry and Web Services Gateway, and the new Web server options. The navigation has been reworked to reduce the number of clicks required to reach most configuration settings.

In addition, the integration of the Tivoli Performance Viewer and IBM HTTP Server management is new.

Integration of the Tivoli Performance Viewer

The Tivol Performance Viewer has been integrated into the WebSphere Administrative Console. Now monitoring and viewing statistics is simply a matter of accessing the Administrative Console. Before viewing graphics metrics, the user is required to install the Adobe SVG Viewer. The monitoring itself is being done from a Web application running in the application server. Before V6, the Tivoli Performance Viewer was a client application that displayed and logged metrics.

Note that there is some additional overhead on a node where you run Tivoli Performance Viewer monitoring. However, this is easily accepted given the simplicity that the new integrated version provides.

System applications

New in WebSphere Application Server V6 is the concept of system applications. System applications are regarded as applications that run in WebSphere, but that do not process client requests. Usually, these applications are used by WebSphere to manage its environment. Two good examples are the *adminconsole* and the *filetransfer* applications.

System applications are installed when WebSphere is initially installed and can only be updated with product fixes or updates. Users cannot change metadata for system applications, such as J2EE bindings or extensions, unless the metadata assigns users and groups for security purposes. Nonsecurity-related metadata requiring a change must be updated through a product fix or upgrade.

System applications are not shown in the list of installed applications on the Administrative Console under the Enterprise application collection Web page. Nor are they shown in wsadmin or Java APIs. This prevents the user from accidently starting, stopping or removing an important system application.

Application management features

A partial application update has been added to the V6 application management. Cluster application update behavior has been enhanced to support the rollout of application updates to cluster members within a cluster.

Fine-grained application update

Fine-grained application update is one of the new features in WebSphere Application Server V6. It allows you to introduce small changes to running applications and have only the impacted parts of the application when it is restarted.

In WebSphere Application Server V5, an application update function was exposed to various administrative tools. The update function took a new EAR file for an application, deployed it on the WebSphere Application Server platform, and performed the necessary deployment steps to replace the existing application configuration with the new one. In order for the application changes to take effect, the application management logic of the update function stopped the running application, replaced the applications files and then restarted the application. Any update that was done via the update function to a running application would cause a stop and start of the application, even for the smallest update. There were two major problems with the V5 update function: a lack of granularity updates and application downtime.

The new fine-grained update feature in WebSphere Application Server V6 is intended to address the shortcomings encountered in V5. You can now apply a single file update and have WebSphere update that same named file on a running application. It is also possible to start and stop portions of an application and preserve configuration settings such as bindings, Classloader settings, and shared libraries.

Updating an application using the new feature can be done by supplying the update process with an entire EAR file or individual components. Updates to single item, such as a JSP or servlet, would be classified as component update. Complete Web modules or EJB JAR files can also be updated with this feature. And the opting to update an application with a zipped file containing multiple application artifacts can be done. This method provides a simple way for updating multiple files in an application.

Rollout application update option

Rollout application update is another new feature that applies to updating applications in a clustered environment. Before in WebSphere Application Server V5, updating an application in a cluster required the cell to be stopped in order for the update to take place.

Now in WebSphere Application Server V6, the rollout update option allows you to sequentially (one at a time) propagate the changed configuration of an application to all members of the cluster on which the application or module is deployed. This allows the cluster to remain active and have the application update roll through the list of cluster members one at a time. Only the cluster member that is being updated will be unavailable.

Note: The update is performed on a node basis. This means that all application servers on a node are stopped for the application update. To provide the high availability (HA) solution for an updated application, you must have at least two nodes.

Replication

There is a new type of replication domain that is based on the new HA framework. A replication domain may consist of one or more application servers that have a replica of data. WebSphere provides transparent support for replicating data to all servers in the replication domain.

By using data replication domains, you do not have to configure local, remote, and alternate replicators. Any replication domains that were created with a previous version of WebSphere Application Server might be multi-broker domains. Existing multi-broker domains remain functional, but after you upgrade your deployment manager, you can create only data replication domains in the administrative console. For improved performance and easier configuration, migrate any existing multi-broker domains to the new data replication domains.

Default message provider

In WebSphere Application Server V5, the embedded messaging server was a stripped down version of WebSphere MQ 5.3. It wasn't integrated with WebSphere Application Server. That implementation had the following limitations:

- Issues with scalability and single point of failure
- No connectivity to external WebSphere MQ
- Security not fully integrated

Now a default messaging provider is installed. It runs as part of WebSphere Application Server V6 and needs no further administration. It is based on service integration technologies.

The default messaging provider supports JMS 1.1 domain-independent interfaces (sometimes referred to as *unified* or *common interfaces*). This enables applications to use the same, common, interfaces for both point-to-point and publish/subscribe messaging. This also enables both point-to-point and publish/subscribe messaging within the same transaction.

This default messaging provider integration with WebSphere Application Server V6 embraces:

- WebSphere Security
- Common install process
- WebSphere system management
- The administrative console provides WebSphere MQ Explorer-type management.
- All Java implementation within the server process (no external processes) that coexist with WebSphere MQ
- Performance monitoring, trace, and problem determination

Other enhancements include:

- Clustering enablement for scalability and high availability
- More flexible Quality of Service (QoS) for message delivery
- Connectivity into a WebSphere MQ network
- Improved performance for in-process messaging

WebSphere configuration archive

Duplicating server configurations is an important task of system management and administration. In the previous release of WebSphere, it was possible to perform such tasks via the wsamdin scripts. Now a new method is available in WebSphere Application Server V6 to save a configuration by exporting or importing WebSphere configurations. This is the *configuration archive* concept. Configuration archiving allows you to create a complete or partial archive of an existing WebSphere Application Server configuration.

Configurations can be virtualized, making them portable to use for creating additional configurations. Virtualizing a configuration removes dependencies, such as a host name, and makes the configuration portable. New configurations can be created or based on configuration archives. Configuration archives may consist of complete configurations or partial configurations.

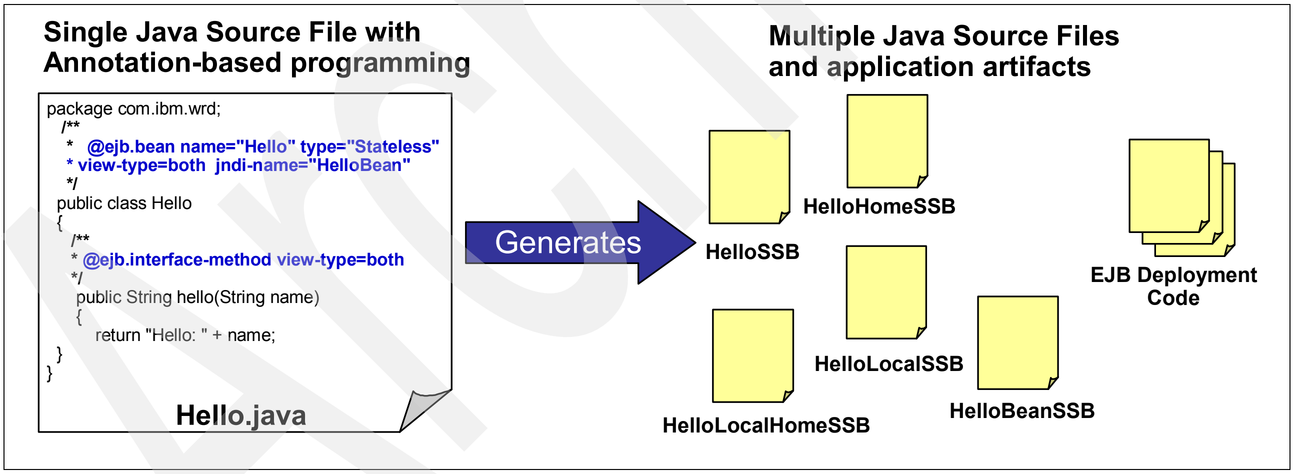
WebSphere Rapid Deployment

WebSphere Rapid Deployment (WRD) speeds and simplifies the development and testing of applications. Its capabilities include annotation-based programming, deployment automation, and change-triggered processing. Since WRD uses existing application server administration features, it does not require any additional configuration or setup.

Annotation-based programming

Annotation-based programming is the notion of allowing the developer to add more metadata into the source code of their application and using that additional metadata to derive the extra artifacts necessary to execute the application in a J2EE environment. The goal of annotation-based programming is to minimize the number of artifacts that the developer has to create and understand, simplifying their development experience. This support is integrated with the Application Server Toolkit (AST) and IBM Rational Application Developer.

Figure A-1 shows the example of annotation-based programming.



```
package com.ibm.wrd;
  /**
   *  @ejb.bean name="Hello" type="Stateless"
   * view-type=both  jndi-name="HelloBean"
   */
  public class Hello
  {
    /**
    * @ejb.interface-method view-type=both
    */
    public String hello(String name)
    {
      return "Hello: " + name;
  }
}
```

Single Java Source File with Annotation-based programming
Hello.java
Generates
Multiple Java Source Files and application artifacts
HelloSSB
HelloHomeSSB
HelloLocalSSB
HelloLocalHomeSSB
HelloBeanSSB
EJB Deployment Code

Figure A-1 Annotation-based programming

Deployment automation

Deployment automation enables the automatic installation of applications and modules onto a running WebSphere Application Server. The server can be remote or local.

Deployment automation is the notion that the system monitors changes being made by the user and automatically ensures that these changes are reflected in a running copy of the application. To do this, the system makes decisions about the default settings necessary to minimize the interaction required by the user. For example, the user can place EARs, application modules (WARs, EJB JARs), or application artifacts (Java source files, Java class files, images, XML, HTML, etc.) into a configurable location on their file system. Then WRD automatically detects the addition or modification of those parts and performs the necessary packaging steps to produce a running application on the WebSphere Application Server. This process is performed in an efficient manner, keeping the steps involved to a minimum.

There are two different forms of WRD deployment automation.

- Automatic installation method

  This method allows the user to export their EAR file modules (WAR, EJB JAR, etc.) and place them in a directory that was configured for automatic application installation (autoappinstall) style in WRD. The placement of the fully-formed J2EE application files is detected by the WRD feature. WRD constructs a J2EE application that can be deployed and installed on WebSphere Application Server. If the file is removed, an uninstall application operation is sent to the WebSphere Application Server.

- Free form method

  This method gives a developer the freedom of not having to understand the structure of a J2EE application. Free form allows the developer to modify a Java artifact and place it into the directory which has been configured to be in free form mode. The addition of the file is detected by the free form WRD feature which then initiates a packaging process that creates a fully formed J2EE application. After construction of the application is complete, the application is deployed and installed on the target WebSphere Application Server.

  Any additions or deletions of files in the monitored directory will cause WRD to send the appropriate command to WebSphere Application Server to perform a specified function to reflect the application update. Free form uses the fine-grained application update for modifying applications on the server. The application WAR will be updated and may not require the application to be restarted.

IBM Service Integration Technologies (messaging)

New in WebSphere Application Server V6 is the Service Integration Technologies infrastructure. This infrastructure replaces the embedded messaging that existed in WebSphere Application Server V5.

The service integration bus provides the communication infrastructure for messaging and service-oriented applications, unifying this support into a common component. The service integration bus is a JMS provider. It is JMS 1.1-compliant for reliable message transport, and it has the capability of intermediary logic to adapt message flow intelligently in the network. Plus it supports the attachment of Web services requestors and providers.

The service integration bus capabilities have been fully integrated within WebSphere Application Server V6. They can take advantage of WebSphere security, administration, performance monitoring, trace capabilities, and problem determination tools.

The service integration bus is often referred to as simply a *bus*. When used to host JMS applications, it is also often referred to as a *messaging bus*.

Figure A-2 illustrates the service integration bus with respect to the enterprise service bus.

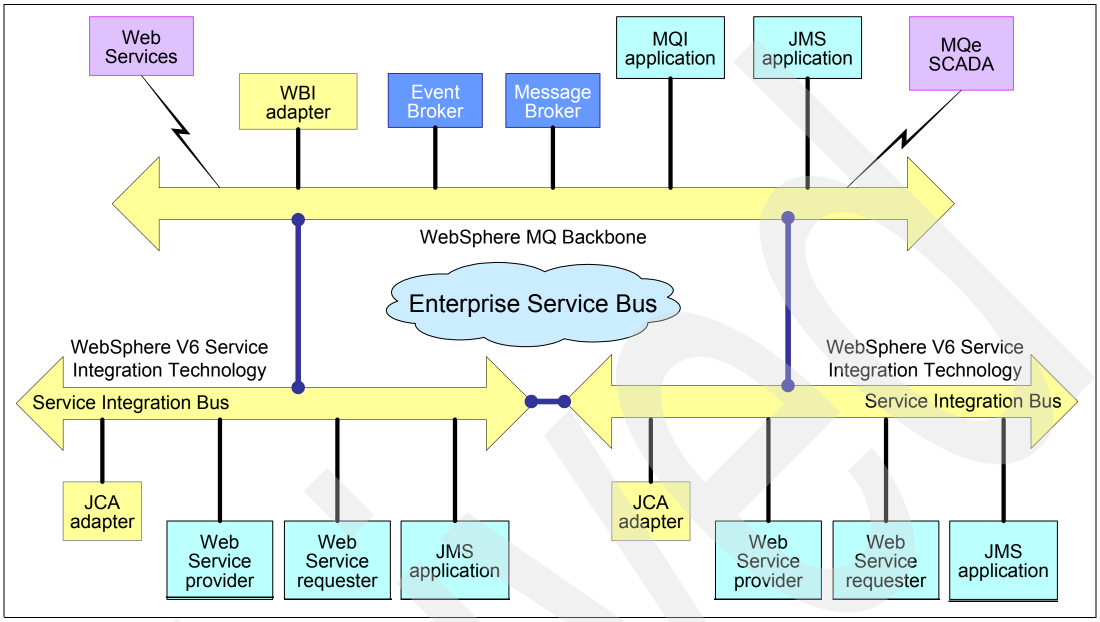


Figure A-2 Service integration bus and the enterprise service bus

A service integration bus consists of the following items:

- Bus members

  These are application servers or clusters that have been added to the bus.

- Messaging engine

  This refers to the application server or cluster component that manages bus resources. When a bus member is defined, a messaging engine is automatically created on the application server or cluster. The messaging engine provides a connection point for clients to produce, or from where to consume, messages.

  An application server has one messaging engine per bus of which it is a member. A cluster has at least one messaging engine per bus and can have more. In this case, the cluster owns the messaging engine or engines and determines on which application server or servers the messaging engines will run.

- Destinations

  This is the location within the bus to which applications attach to exchange messages. Destinations can represent Web service endpoints, messaging point-to-point queues, or messaging publish/subscribe topics. Destinations are created on a bus and hosted on a messaging engine.

- Message store

  Each messaging engine uses a set of tables in a data store (JDBC database) to hold information such as messages, subscription information, and transaction states. Messaging engines can share a database, each using its own set of tables. The message store can be backed by any JDBC database supported by WebSphere Application Server.

Application support

The service integration bus supports the following application attachments:

- Web services
- Requestors using the JAX-RPC API
- Providers running in WebSphere Application Server as stateless session beans and servlets (JSR-109)
- Requestors or providers attaching via SOAP/HTTP or SOAP/JMS
- Messaging applications with inbound messaging using JFAP-TCP/IP (or wrapped in Secure Sockets Layer (SSL) for secure messaging)

  JFAP is a proprietary format and protocol used for service integration bus messaging providers.
- MQ application in an MQ network using MQ channel protocol
- JMS applications in WebSphere Application Server V5 using WebSphere MQ client protocol
- JMS applications in WebSphere Application Server V6

Service integration bus and messaging

With WebSphere Application Server Express or Base, you typically have one stand-alone server with one messaging engine on one service integration bus. With Network Deployment, however, you have more flexibility by using a service integration bus and messaging.

The following topologies are valid:

- One bus and one messaging engine (application server or cluster)
- One bus with multiple messaging engines
- Multiple buses within a cell which may or may not be connected to each other
- Buses connected between cells
- One application server that is a member of multiple buses and that has one messaging engine per bus
- A connection between a bus and a WebSphere message queue manager

  When using this type of topology, consider the following points:

  – WebSphere message queue can coexist on the same machine as the WebSphere default messaging provider. In V5, the embedded JMS server and WebSphere MQ could not coexist on the same machine.

  – A messaging engine cannot participate in a WebSphere MQ cluster.

  – You can configure the messaging engine to look like another queue manager to WebSphere MQ.

  – WebSphere applications can send messages directly to WebSphere MQ or through the service integration bus.

  – You can have multiple connections to WebSphere MQ, but each connection must be to a different queue manager.

  – The WebSphere Application Server V5 JMS client can connect to V6 destinations. Also, a V6 JMS application can connect to an embedded messaging provider in a V5 server if one is configured. However, you cannot connect a V5 embedded JMS server to a V6 bus.

Mediation

Mediation manipulates a message as it traverses the messaging bus (destination). For example, mediation can perform the following actions:

- Transform the message
- Reroute the message
- Copy and route the message to additional destinations
- Interact with non-messaging resource managers (for example, databases)

You control mediation using a mediation handler list. The list is a collection of Java programs that perform the function of a mediation that are invoked in sequence.

Clustering

In a distributed server environment, you can use clustering, by adding a cluster as a bus member to achieve high availability and scalability.

- High availability

  One messaging engine is active in the cluster. In the event that the cluster member hosting the messaging engine fails, the messaging engine is restarted by a peer member after 20 seconds. In the event that the cluster member hosting the messaging engine fails, the messaging engine is restarted by a peer member after 20 seconds.

- Scalability

  A single messaging destination can be partitioned across multiple active messaging engines in the cluster. The messaging order is not preserved.

Quality of Service

You can define QoS on a destination basis to determine how messages are (or are not) persisted. You can also specify QoS within the application.

Message Driven Beans

With EJB 2.1, Message Driven Beans (MDB) are the Java components that listen to queues and topics. They are linked to the appropriate destinations on the service integration bus using JCA connectors (ActivationSpec objects). Support is also included for EJB 2.0 MDBs to be deployed against a listener port.

Common networking services

The new channel framework model provides a common networking service for all components, including IBM service integration technologies, WebSphere Secure Caching Proxy, and the High Availability Manager (HAManager) core group bridge service. This model consists of network protocol stacks or transport chains that are used for I/O operations within an application server environment.

Transport chains consist of one or more types of channels, each of which supports a different type of I/O protocol, such as TCP, DCS, or HTTP. Network ports can be shared among all of the channels within a chain. The channel framework function automatically distributes a request arriving on that port to the correct I/O protocol channel for processing.

Transport channel service

The way in which a transport is implemented in WebSphere has been updated for version 6. A new framework for creating and facilitating transports has been implemented which makes common network transport services available to all components. This new transport service makes it easy and efficient to add protocol support. It is a pluggable architecture that uses adapters to handle communications with the application server. The new transport channel service takes advantage of the Java 1.4 non-blocking I/O feature.

Some benefits of the new transport channel service are:

- Scalability improvements
- TCP port sharing
- Configurable thread pools

High availability

WebSphere Application Server uses a HAManager to eliminate single points of failure. A HAManager is responsible for running key services on available application servers rather than on a dedicated one, such as the deployment manager. The HAManager also provides peer-to-peer failover for critical services by always maintaining a backup for these services.

A HAManager continually monitors the application server environment. If an application server component fails, the HAManager takes over the in-flight and in-doubt work for the failed server. This action significantly improves application server availability.

In a highly available environment, all single points of failure are eliminated.

> **Important:** Make sure that you understand that the HAManager eliminates a single point of failure within the WebSphere network of components. Any components outside of WebSphere configuration must be protected separately.

Because the HAManager function is dynamic, any configuration changes that you make and save while an application server is running are eventually picked up and used. You do not have to restart an application server to enable a change.

For example, you can change a policy for a messaging engine high availability group while the messaging engine is running. Then the new policy is dynamically loaded and applied, and the behavior of the messaging engine reflects this change.

A HAManager focuses on recovery support and scalability in the following areas:

- Messaging
- Transaction managers
- Workload Management (WLM) controllers
- Application servers
- WebSphere partitioning facility instances

To provide this focused failover service, the HAManager supervises the JVMs of the application servers that are core group members. The HAManager uses one of the following methods to detect failures:

- An application server is marked as failed if the socket fails.

  This method uses the KEEP_ALIVE function of TCP/IP. It is tolerant of extreme application server loading, which might occur if the application server is swapping or thrashing heavily.

We recommend that you use this method to determine a JVM failure if you are using multicast emulation and are running enough JVMs on a single application server to push the application server into extreme CPU starvation or memory starvation.

- A JVM is marked as failed if it stops sending heartbeats for a specified time interval.

  This method is referred to as *active failure detection*. When it is used, a JVM sends one heartbeat or a pulse every second. If the JVM is unresponsive for more than 20 seconds, it is considered down.

  You can use this method with multicast emulation. However, you must use it for true multicast addressing.

In either case, if a JVM fails, the application server on which it is running is separated from the core group. Any services running on that application server are failed over to the surviving core group members.

A JVM can be a node agent, an application server, or a deployment manager. If a JVM fails, any WebSphere service running in that JVM is restarted on a peer JVM after the failure is detected. This peer JVM is already running and eliminates the normal startup time, which potentially can be minutes.

All of the application servers in a cell are defined as members of a core group. Each core group has only one logical HAManager that services all of the members of that core group. The HAManager is responsible for making the services within a core group highly available and scalable. It continually polls all of the core group members to verify that they are active and healthy.

A policy matching program is used to localize certain policy-driven components and to place these components into high availability groups. When a core group member fails, the HAManager assigns the failing member’s work to the same type of component from the same high availability group.

WebSphere Application Server provides a default core group that is created during installation. New server instances are added to the default core group as they are created. The WebSphere Application Server environment can support multiple core groups, but one core group is usually sufficient for most environments.

A HAManager is comprised of a variety of components. All of the components in a HAManager infrastructure work together to ensure that peer-to-peer failover is effectively protecting the application server environment from failures. All of these components must be active and properly configured to achieve a highly available infrastructure.

Clustering enhancements

There are several major enhancements in clustering support.

- Change in the clustering architecture
- Rewrite of the Data Replication Service code
- Support for the state replication of the Stateful Session EJBs

Unified clustering

Prior to the Unified Cluster Framework, the routing of each protocol was done by separate groups based on specific requirements for a given protocol (cluster). The logic for HTTP servers, Web Services Gateway, and EJB servers was all unique.

The Unified Cluster Framework standardizes how the cluster data is collected, propagated, and routed using a standard consistent architecture. As new technologies are introduced into

WebSphere, they can also take advantage of the framework. For instance, the HTTP server's plug-in logic is different from the EJB (IIOP) routing logic, which is different than the proxy. The quality of the code has been improved since the logic is based on a single consistent architecture instead of several specific ones.

The Unified Cluster Framework provides a more consistent way to view, configure, and administer the different types of clusters (JMS, HTTP, Web Services Gateway, IIOP) since these clusters use the same architecture. Clustered protocols can now take advantage of the HAManager, which improves the QoS.

The following list summarizes the clustering enhancements:

- Edge product integration

  The Edge function, which was separate in WebSphere Application Server V5, is integrated in WebSphere Application Server V6 with the support of unified clusters.

- Operational ease of use

  The view and use of clusters will be administered in a unified and consistent manner for all protocols.

- Third-party integration

  This feature allows customers to have better control of a cluster definition when they need it and still fit into the consistent WebSphere view.

- Continuous availability

  This allows clients to build the infrastructure needed to enable a better story for doing version upgrades on continuous running systems.

- High availability

  HA makes Workload Manager (WLM) a highly available service (using the HAManager), which makes cluster and routing information always available.

- Consistency

  The new WLM functions (weighted distribution, Enterprise Workload Manager (eWLM) integration, service-level agreement (SLA), hardware provisioning, and so on) are implemented once for all protocols.

- Central control of Web Container clusters
- Backup server lists for EJBs
- Real-time updates of the HTTP route table
- Support for future capabilities such as queuing, classification, and work prioritization in enhanced products, like WebSphere Application Server Extended Deployment (XD).

Data replication service enhancements

There are a number of enhancements in WebSphere Application Server V6 related to the data replication service. One of the major improvements is integration with the HAManager, which provides the following benefits:

- Improves performance and scalability
  - Provides a more optimized communication stack
  - Allows for use of both unicast and multicast IP
  - Improves replication speed in the range of four to eight times

- Improves high availability and failure recovery
  - It leverages failure detection provided by high availability services.
  - Along with the WLM and Unified Clustering integration, this allows for *active failure recovery*. For example, with HTTP Session replication, if the affinity server for an HTTP Session goes down, WLM can route to another server that has a backup copy ready to use.
- Improves usability by leveraging group services to simplify partitioning. Data Replication Service now supports *n-replica*, where a customer simply defines the number of backup copies they want.

Support for failover of Stateful Session EJBs

New in WebSphere Application Server V6 is the support for failover of Stateful Session EJBs. This support relies upon the Data Replication Service and Workload Management services.

Security

There have been few enchancements in WebSphere Application Server V6 security functions. Most of the security-related functions and administration are similar to version 5. New to version 6 is the support for J2EE 1.4 Java Authorization Contract with Containers (JACC).

In version 6 security, all WebSphere editions use the same security capabilities. This was not the case in version 5, where WebSphere Express had limited security functions and did not have the Lightweight Third Party Authentication (LTPA) mechanism.

The biggest improvement in WebSphere Application Server V6 security involves the following specifications:

- JACC provider support
- Java 2 security manager
- JCA 1.5 support
- SSL channel framework
- Web authentication
- Web services security

Java Authorization Contract with Containers 1.0 support

JACC 1.0 support details the contract requirements for J2EE containers and authorization providers. With such details, authorization providers can make access decisions for resources in J2EE 1.4 application servers such as WebSphere Application Server. This support facilitates the plug-in of third-party authorization servers. When you use this feature, WebSphere Application Server supports Tivoli Access Manager as the default JACC provider.

Java 2 security manager

WebSphere Application Server V6 provides you with greater control over the permission granted to applications for manipulating non-system threads. You can permit applications to manipulate non-system threads by using the was.policy file. By default, the thread control permissions are disabled.

JCA 1.5 support

WebSphere Application Server V6 supports the JCA Version 1.5 specification, which provides new features such as the inbound resource adapter. From a security perspective, WebSphere Application Server V6 provides an enhanced custom principal and credential programming interface and custom mapping properties at the resource reference level.

SSL channel framework

The SSL channel framework incorporates the new IBMJSSE2 implementation. It also separates the security function of Java Secure Sockets Extension (JSSE) from the network communications function.

Web authentication

WebSphere Application Server V6 enables you to use the Java Authentication and Authorization Service (JAAS) programming model to perform Web authentication in your application code. To use this function, you must create your own JAAS login configuration by cloning the WEB_INBOUND login configuration and defining a “cookie=true” login option.

After a successful login, using your login configuration, the Web login session is tracked by single signon (SSO) token cookies. This option replaces the SSOAuthenticator interface, which was deprecated in WebSphere Application Server V4.

Web services security

WebSphere Application Server V6 increases the extensibility of Web services security by providing a pluggable architecture. The implementation in version 6 includes many of the features described in the Organization for the Advancement of Structured Information Standards (OASIS) Web Services Security Version 1 standard. As part of this standard, WebSphere Application supports:

- Custom, pluggable tokens that are used for signing and encryption
- Pluggable signing and encryption algorithms
- Pluggable key locators for locating a key that is used for digital signature or encryption
- Signing or encrypting elements in a SOAP message
- Specifying the order or the signing or encryption processes

B

Installing WebSphere Application Server Version 6

This appendix covers the prerequisites for installing WebSphere Application Server on iSeries. It also presents the different methods for you to install WebSphere Application Server V6 on your iSeries server. And it explains how to verify the installation and apply fixes to WebSphere Application Server.

Installing WebSphere Application Server V6 is a prerequisite for setting up and configuring either of the scenarios (small company or large company) presented in this redbook. Each scenario uses different network components in the topology. Depending on the function of a component, you need to install some or all features of WebSphere Application Server. Since each scenario details the features that are needed, you can use this appendix as a reference to install the necessary software.

iSeries prerequisites

Prior to installing WebSphere Application Server V6, certain software products are required to be installed on your iSeries system. The following is only a *partial* list:

- OS/400 Version 5 Release 2 (V5R2) or Version 5 Release 3 (V5R3)
- IBM Developer Kit for Java Version 1.4 (5722-JV1 option 6)
- OS/400 Qshell (5722-SS1 option 30)
- OS/400 Host Servers (5722-SS1 option 12)
- Cryptographic Access Provider 128-bit for iSeries (5722-AC3)

> **Attention:** For a detailed listing of the required iSeries software and hardware, refer to the WebSphere Information Center at:
>
> `http://publib.boulder.ibm.com/infocenter/wsdoc400/index.jsp?topic=/com.ibm.websphere.iseries.doc/info/ae/ae/prqsvr.htm`

What to install

Depending on the topology that you will implement for your business, you may need to install the *Base option plus one other option of the WebSphere Application Server product.

For Version 6, you may see a combination of the options in Table B-1 in the Display Software Resources (DSPSFWRSC) panel on the iSeries.

Table B-1 Available options of WebSphere Application Server V6

Resource ID	Option	Description
5733-W60	*BASE	WebSphere Application Server for OS/400 V6
5733-W60	1	WebSphere Application Server V6 Express
5733-W60	2	WebSphere Application Server V6 (Base)
5733-W60	3	WebSphere Application Server V6 Network Deployment

Only the combinations in Table B-2 of the *Base option and other options can exist on an iSeries system.

Table B-2 Available combinations of installed WebSphere Application Server options on iSeries

Option	1	2	3
*Base[a]	✗	✗	✗
*Base	✓	✗	✗
*Base	✓	✗	✓
*Base	✗	✓	✗
*Base	✗	✓	✓
*Base	✗	✗	✓

[a] The *Base option may exist alone on an iSeries if one or both of the Web server plug-ins or Application Client features are installed.

The *Base option *always* exists on the system alone or with one or two other options.

Installation methods

Unlike the past releases of WebSphere Application Server, V6 is no longer packaged as an OS/400 licensed program product (LPP). A product definition file is loaded on the system to allow the user to view the options of WebSphere Application Server that are loaded by using the GO LICPGM or DSPSFWRSC OS/400 commands.

The following installation methods are available to the user:

- GUI installation from a remote workstation: See “GUI installation from a remote workstation” on page 257.
- Command line installation from a remote workstation (Silent method): See “Command line installation from a remote workstation” on page 263.
- Command line installation from the iSeries (Silent method): See “Command line installation from the iSeries” on page 264.

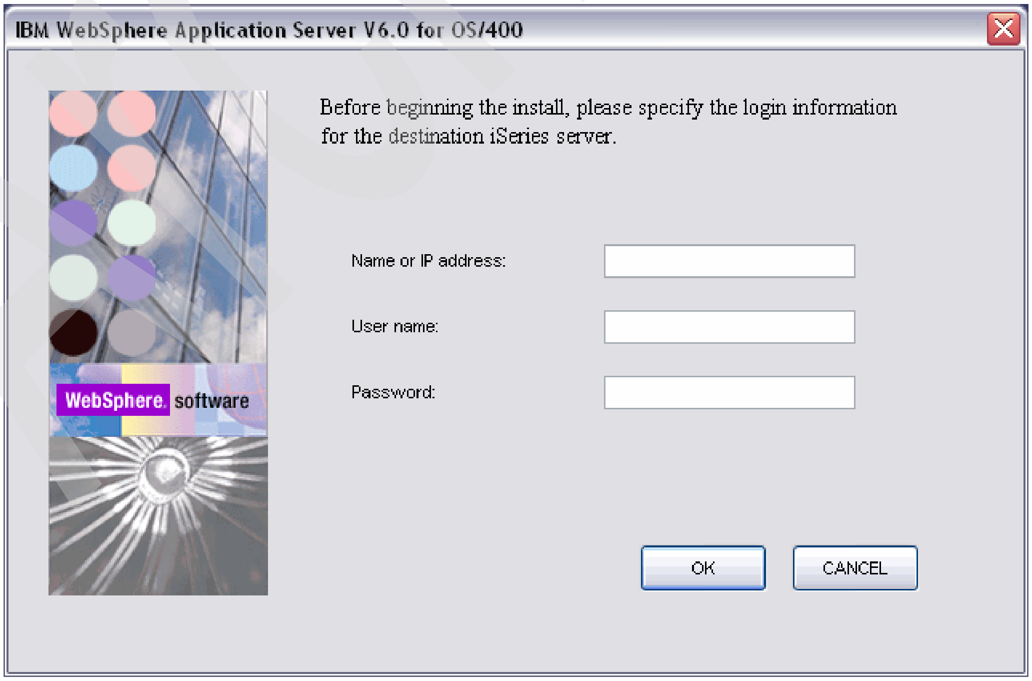
GUI installation from a remote workstation

The workstation that is used for the installation must have a CD-ROM or DVD drive and be connected on the same TCP/IP network as your iSeries. Complete the following steps:

1. Start the host servers on the iSeries, using the command:

   ```
   STRHOSTSVR *ALL
   ```

2. Place the WebSphere Application Server V6 CD in the CD-ROM of the workstation.
3. The InstallShield program should start automatically. If it does not start automatically, the program can be initiated by double-clicking the **install.exe** file, found in the WAS directory, on the CD.
4. As shown in Figure B-1, you see the initial window where the logon information is entered. Enter the name or IP address of the iSeries system. Then enter a user ID with *SECOFR and *SECADM authorities and the user’s password. Click **OK** to continue the installation.

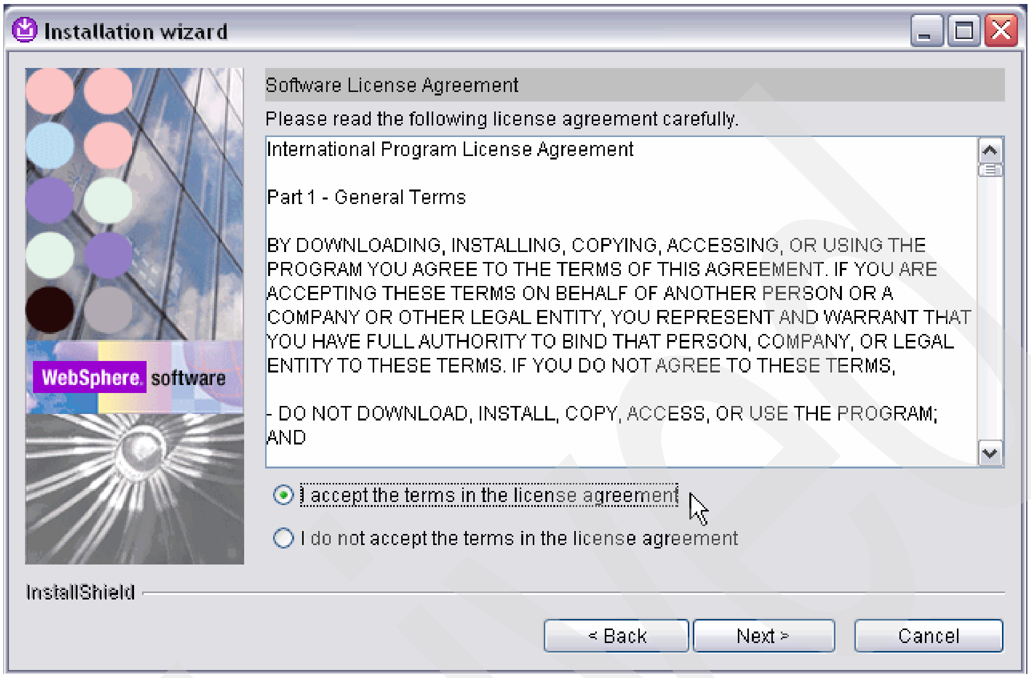


IBM WebSphere Application Server V6.0 for OS/400

Before beginning the install, please specify the login information for the destination iSeries server.

Figure B-1 Logon information window

5. The Software License Agreement panel (Figure B-2) opens. Review the agreement and specify whether you accept the agreement. Click **Next**.



Software License Agreement

Please read the following license agreement carefully.

International Program License Agreement

Part 1 - General Terms

BY DOWNLOADING, INSTALLING, COPYING, ACCESSING, OR USING THE PROGRAM YOU AGREE TO THE TERMS OF THIS AGREEMENT. IF YOU ARE ACCEPTING THESE TERMS ON BEHALF OF ANOTHER PERSON OR A COMPANY OR OTHER LEGAL ENTITY, YOU REPRESENT AND WARRANT THAT YOU HAVE FULL AUTHORITY TO BIND THAT PERSON, COMPANY, OR LEGAL ENTITY TO THESE TERMS. IF YOU DO NOT AGREE TO THESE TERMS,

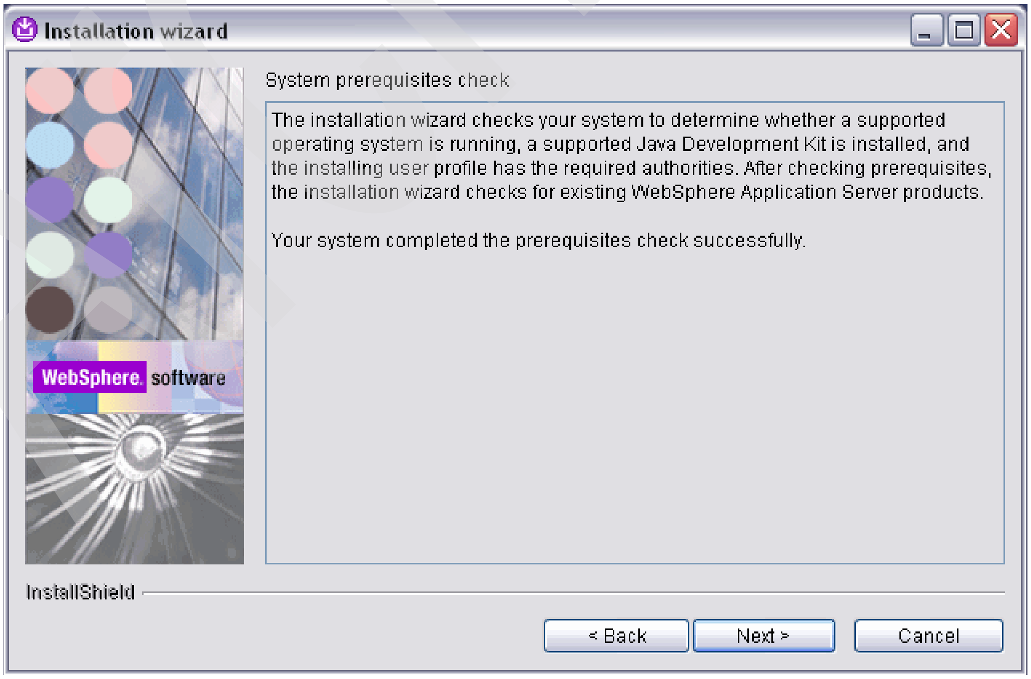
- DO NOT DOWNLOAD, INSTALL, COPY, ACCESS, OR USE THE PROGRAM; AND

● I accept the terms in the license agreement

○ I do not accept the terms in the license agreement

Figure B-2 Software License Agreement panel

6. The System prerequisite check panel (Figure B-3) opens. It shows the result of a successful system prerequisites check. If the prerequisite check fails, review the messages and correct the problem. For an example of a prerequisite failure, see "Prerequisite check failures" on page 270. Click **Next**.



System prerequisites check

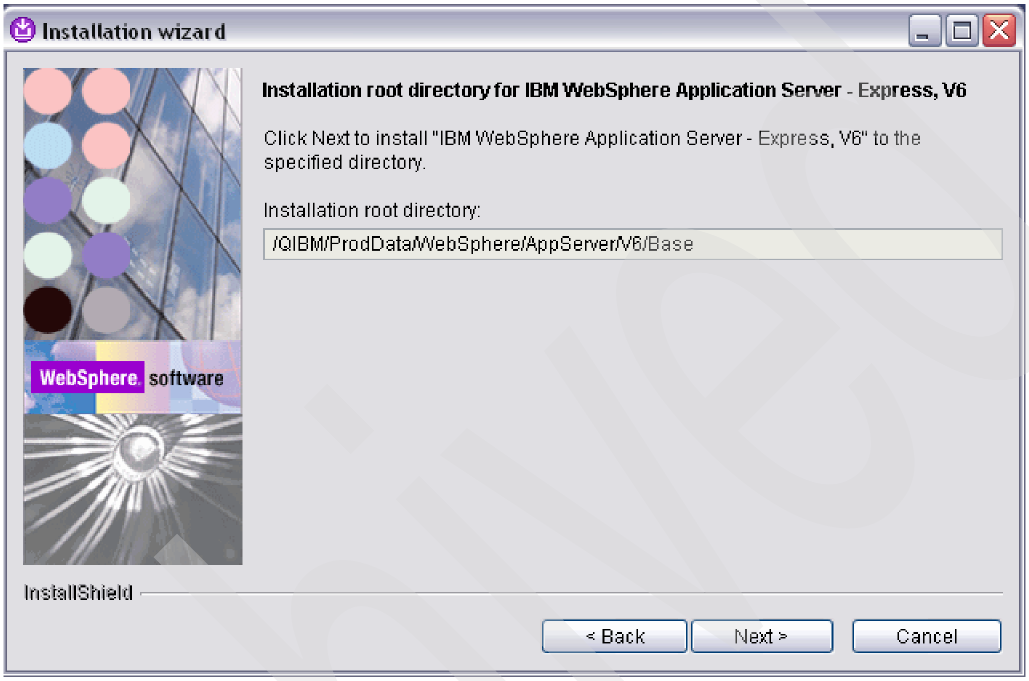
The installation wizard checks your system to determine whether a supported operating system is running, a supported Java Development Kit is installed, and the installing user profile has the required authorities. After checking prerequisites, the installation wizard checks for existing WebSphere Application Server products.

Your system completed the prerequisites check successfully.

Figure B-3 System prerequisite check panel

7. In the Installation root directory panel (Figure B-4), you see the installation root directory for a WebSphere Application Server V6 Express or WebSphere Application Server V6 (Base) installation. If WebSphere Application Server V6 Network Deployment is being installed, the installation root directory is /QIBM/ProdData/WebSphere/AppServer/V6/ND.

   In either case, the directory is created by the installation program if it does not exist on the iSeries. The user cannot change the installation directory. Click **Next**.



Installation root directory for IBM WebSphere Application Server - Express, V6

Click Next to install "IBM WebSphere Application Server - Express, V6" to the specified directory.

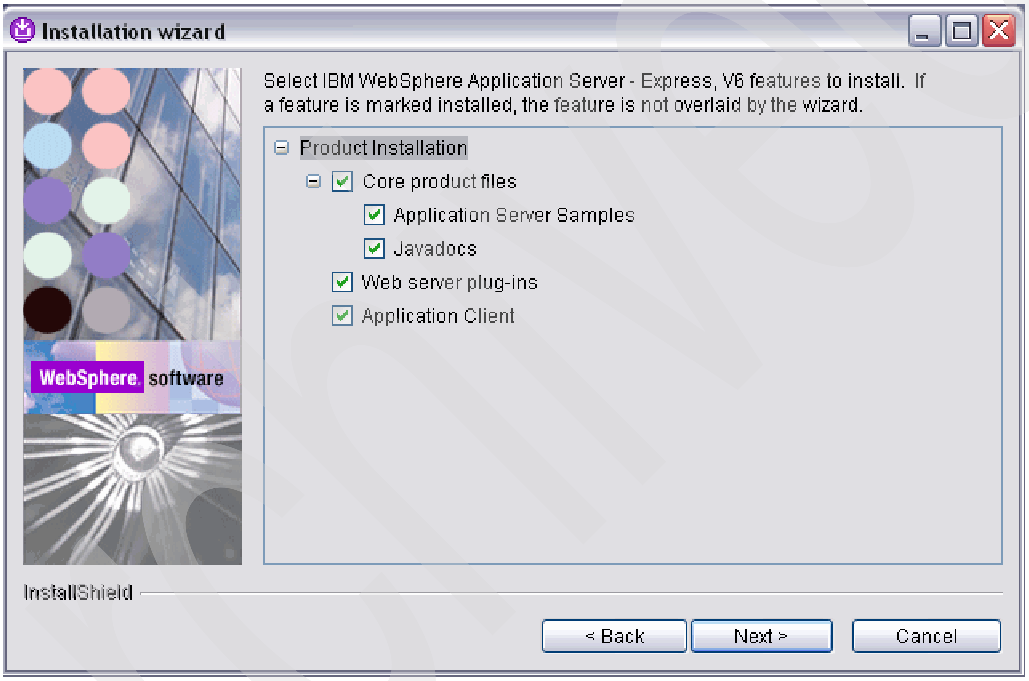
Installation root directory:

/QIBM/ProdData/WebSphere/AppServer/V6/Base

Figure B-4 Installation root directory panel

8. In the Select IBM WebSphere Application Server Express features panel (Figure B-5), you see the selection of features that you can select for installation. By default, all features and sub-features are selected. On this window, you can either accept all selections or choose the specific features that are required on the system. Then click **Next**.

> **Note:** There are several combinations for installation options.
>
> - If the Core Product Files feature is selected, the Web server plug-ins and Application Client features are automatically selected.
> - Application Server Samples is a sub-feature. You cannot install it without installing the Core Product Files feature.
> - Javadocs is a sub-feature. You cannot install it without installing the Core Product Files feature.

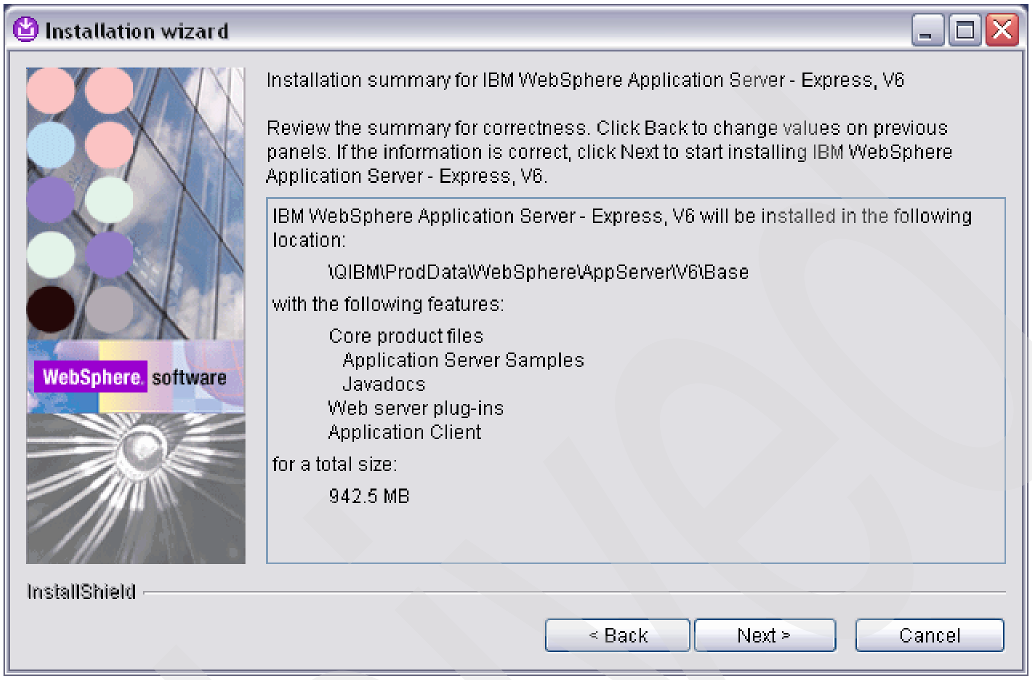


Select IBM WebSphere Application Server - Express, V6 features to install. If a feature is marked installed, the feature is not overlaid by the wizard.

⊟ Product Installation
- ⊟ ☑ Core product files
  - ☑ Application Server Samples
  - ☑ Javadocs
- ☑ Web server plug-ins
- ☑ Application Client

Figure B-5 WebSphere Application Server features panel

9. In the Installation summary panel (Figure B-6), you see which WebSphere Application Server features will be installed on the iSeries and how much disk space is necessary for those features. Click **Next** to start the installation.



Installation summary for IBM WebSphere Application Server - Express, V6

Review the summary for correctness. Click Back to change values on previous panels. If the information is correct, click Next to start installing IBM WebSphere Application Server - Express, V6.

IBM WebSphere Application Server - Express, V6 will be installed in the following location:

\QIBM\ProdData\WebSphere\AppServer\V6\Base

with the following features:

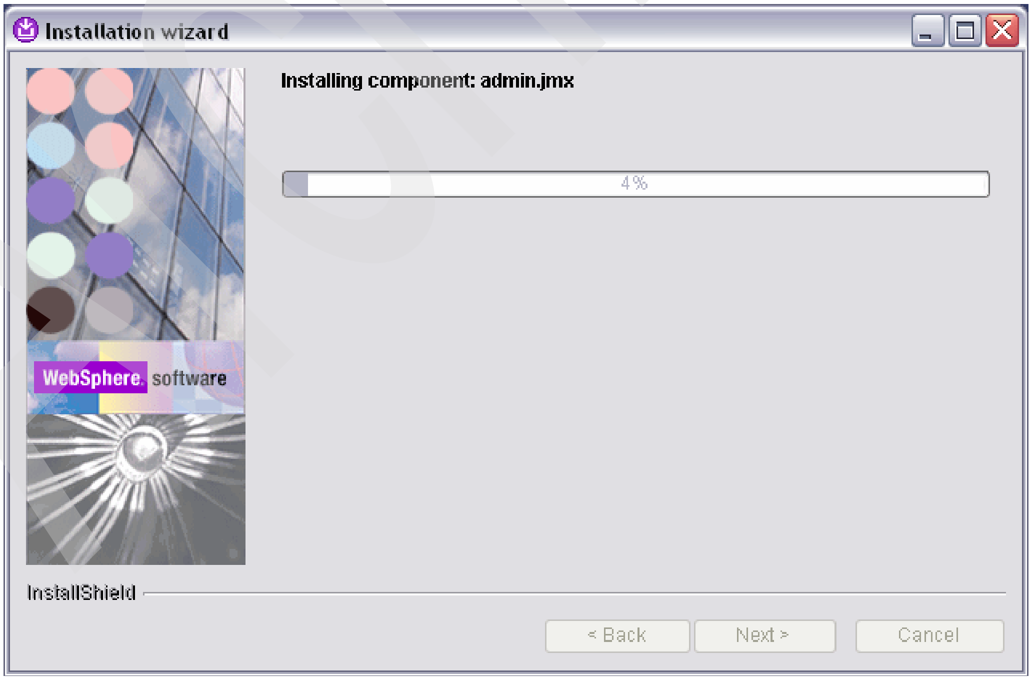
Core product files
Application Server Samples
Javadocs
Web server plug-ins
Application Client

for a total size:

942.5 MB

Figure B-6 Installation summary panel

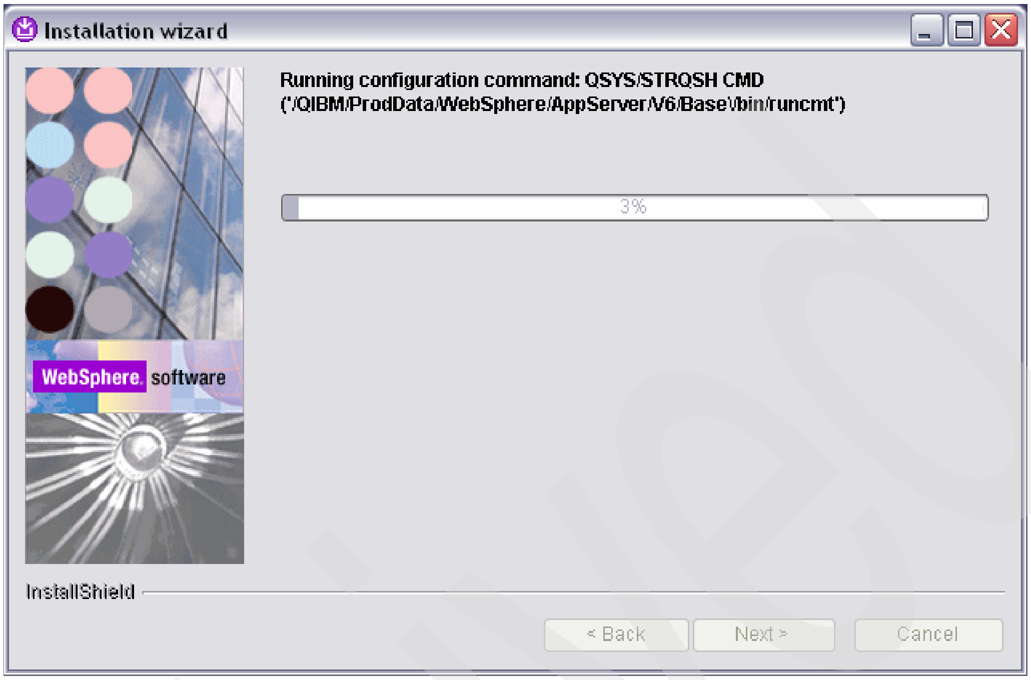
10. You see the progress of the WebSphere Application Server features installation as shown in the example in Figure B-7.



Installing component: admin.jmx

Figure B-7 Installation progress panel

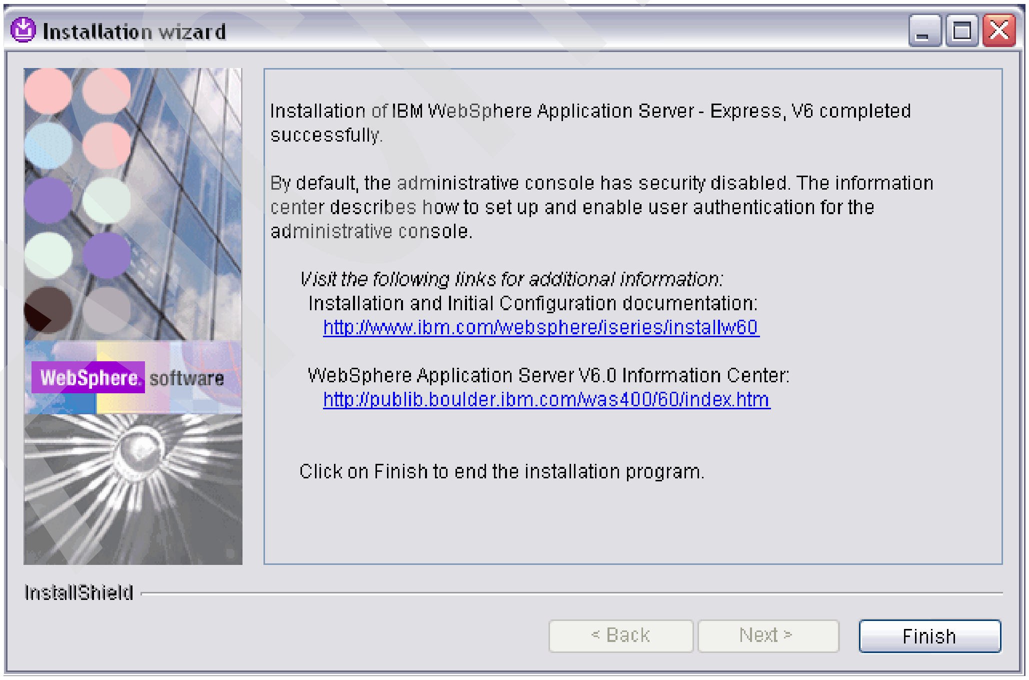
11. After the installation of WebSphere Application Server completes, you see a new panel (Figure B-8), showing the progress of the default profile creation.



Running configuration command: QSYS/STRQSH CMD ('/QIBM/ProdData/WebSphere/AppServer/V6/Base/bin/runcmt')

Figure B-8 Profile creation progress panel

12. When the profile creation is completed, you see the installation summary panel (Figure B-9). Review the information in the window and click **Finish** to exit the Installation wizard.



Installation of IBM WebSphere Application Server - Express, V6 completed successfully.

By default, the administrative console has security disabled. The information center describes how to set up and enable user authentication for the administrative console.

Visit the following links for additional information:
Installation and Initial Configuration documentation:
http://www.ibm.com/websphere/iseries/installw60

WebSphere Application Server V6.0 Information Center:
http://publib.boulder.ibm.com/was400/60/index.htm

Click on Finish to end the installation program.

Figure B-9 Installation complete summary panel

To ensure that the installation and profile creation are successful, refer to “Verifying the installation of WebSphere Application Server V6” on page 265.

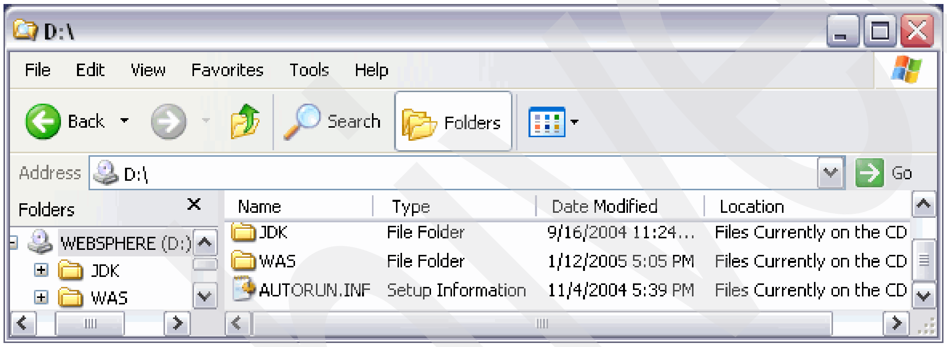
Command line installation from a remote workstation

The workstation that is used for the installation must have a CD-ROM or DVD drive. It must also be connected on the same TCP/IP network as your iSeries. Complete the following steps:

1. Start the host servers on the iSeries.

   ```
   STRHOSTSVR *ALL
   ```

2. Place the WebSphere Application Server V6 CD in the CD-ROM of the workstation.
3. The InstallShield program should start automatically. After the program starts, click **Cancel** to exit.
4. Navigate to the contents of the CD-ROM, as in the example shown in Figure B-10.



Name	Type	Date Modified	Location
JDK	File Folder	9/16/2004 11:24…	Files Currently on the CD
WAS	File Folder	1/12/2005 5:05 PM	Files Currently on the CD
AUTORUN.INF	Setup Information	11/4/2004 5:39 PM	Files Currently on the CD

Figure B-10 Contents of WebSphere Application Server CD

5. In the WAS directory, locate the RESPONSEFILE file. Copy the RESPONSEFILE to the workstation.
6. If you have not already done so, read the IBM International Program License Agreement located in the CD-ROM directory WAS\docs\lafiles. If you agree to the terms of the agreement, proceed to the next step.
7. Edit the RESPONSEFILE. Find the line:

   ```
   -W silentInstallLicenseAcceptance.value="false"
   ```

   Change the value “false” to “true”. Save your changes.

8. Decide which WebSphere Application Server features will be installed. If all features are to be installed, proceed to step 9. If only certain features are to be installed, edit the following lines in the RESPONSEFILE:

   ```
   -P coreRuntimeProductFeatureBean.active="true"
   -P samplesProductFeatureBean.active="true"
   -P javadocsProductFeatureBean.active="true"
   -P webServerPluginFeatureBean.active="true"
   -P applicationClientFeatureBean.active="true"
   ```

   Change the value from “true” to “false” for any feature that does not need to be installed. Save your changes.

9. From a command line, navigate to the WAS directory in the CD-ROM directory.
10. Type the following command:

    ```
    install.exe systemname username password -options responsefile
    ```

Here, *systemname* is the name of the iSeries server, and *username* is an OS/400 user that has *SECOFR and *SECADM authorities. Also, *password* is the associated password of the user, and *responsefile* is the location of the RESPONSEFILE that was changed in step 7 on page 263. In this example, we write the command as:

```
install.exe systemA qsecofr pa33w0rd -options C:\tempdir\RESPONSEFILE
```

Note: The password will be visible when initiating the install.exe from a command line.

If you encounter any problems during the installation, refer to “WebSphere Application Server installation problems” on page 268. Also, because there is no progress indication in this type of installation, refer to “Verifying the installation of WebSphere Application Server V6” on page 265 to ensure that the WebSphere Application Server product is installed correctly and the default profile is created.

Command line installation from the iSeries

Ensure that the iSeries that is used for the installation has a CD-ROM or DVD drive. Complete the following steps:

1. Start the host servers on the iSeries, using this command:

   ```
   STRHOSTSVR *ALL
   ```

2. Place the WebSphere Application Server V6 CD in the CD-ROM or DVD drive of the iSeries system.
3. Sign on to the iSeries with a user that has *SECOFR and *SECADM authority.
4. Use the Copy File OS/400 command to create a copy of the RESPONSEFILE from the CD-ROM directory, for example:

   ```
   CPY OBJ(‘/QOPT/WEBSPHERE/WAS/RESPONSEFILE’) TOOBJ(‘/QIBM/RSPFILE’)
   ```

5. If you have not already done so, read the IBM International Program License Agreement located in the CD-ROM directory /QOPT/WEBSPHERE/WAS/DOCS/lafiles. If you agree to the terms of agreement, proceed to the next step.
6. Edit the RESPONSEFILE that has been copied. Find the line:

   ```
   -W silentInstallLicenseAcceptance.value="false"
   ```

   Change the value “false” to “true”. Save the changes.

7. Decide which WebSphere Application Server features will be installed. If all features are to be installed, proceed to step 8. If only certain features are to be installed, edit the following lines in the RESPONSEFILE:

   ```
   -P coreRuntimeProductFeatureBean.active="true"
   -P samplesProductFeatureBean.active="true"
   -P javadocsProductFeatureBean.active="true"
   -P webServerPluginFeatureBean.active="true"
   -P applicationClientFeatureBean.active="true"
   ```

   Change the value from “true” to “false” for any feature that does not need to be installed. Save your changes.

8. Start a Qshell session and enter the following command:

   ```
   cd /QOPT/WEBSPHERE/WAS
   ```

9. Enter the following command:

   ```
   setup -options responsefile
   ```

   Here, *responsefile* is the fully-qualified path of the location of the responsefile that is changed in step 6.

10. The installation begins, and messages are displayed to indicate the progress. When the setup is complete, the last message line should contain one of the following strings:

    ```
    Installation of IBM WebSphere Application Server - Express, V6 completed successfully.
    Installation of IBM WebSphere Application Server, V6 completed successfully.
    Installation of IBM WebSphere Application Server Network Deployment, V6 completed
    successfully.
    ```

If you encounter any problems during the installation, refer to “WebSphere Application Server installation problems” on page 268. Also refer to “Verifying the installation of WebSphere Application Server V6” on page 265.

Verifying the installation of WebSphere Application Server V6

Complete the following actions to ensure that WebSphere Application Server V6 has installed successfully on the iSeries.

Log.txt file

Review the log file located in the /QIBM/ProdData/WebSphere/AppServer/V6/*edition*/logs/log.txt directory, where *edition* is one of the following types:

- *Base* represents WebSphere Application Server Express or WebSphere Application Server (Base)
- *ND* represents WebSphere Application Server Network Deployment

In the log.txt file, depending on the option that is installed, search for one of the following messages:

```
Installation of IBM WebSphere Application Server - Express, V6 completed successfully.
Installation of IBM WebSphere Application Server, V6 completed successfully.
Installation of IBM WebSphere Application Server Network Deployment, V6 completed
successfully.
```

The message is usually at or near the end of the log.txt file.

DSPSFWRSC

Run the OS/400 DSPSFWRSC command. Scroll through the listing and look for entries like those shown in Table B-1 on page 256.

Alternatively, type the GO LIGPGM OS/400 command. Select option 10 from the Displayed Installed Licensed Programs panel. Scroll through the listing and look for entries similar to those shown in Table B-1 on page 256.

Integrated file system directories

The following Integrated File System (IFS) directories are created when WebSphere Application Server is installed:

- For WebSphere Application Server V6 Express or WebSphere Application Server V6 Base
  - /QIBM/ProdData/WebSphere/AppServer/V6/Base
  - /QIBM/UserData/WebSphere/AppServer/V6/Base

- For WebSphere Application Server V6 Network Deployment
  - /QIBM/ProdData/WebSphere/AppServer/V6/ND
  - /QIBM/UserData/WebSphere/AppServer/V6/ND

Note: All four directories may exist on the iSeries system, depending on the WebSphere Application Server editions that have been installed on the system.

QWAS6 library

The installation of WebSphere Application Server V6 on iSeries creates the QWAS6 library. This library contains a number of object types, including:

- *PGM
- *SRVPGM
- *JOBD
- *FILE
- *JOBQ
- *CLS
- *SBSD
- *PRDDFN
- *PRDLOD

Ensure that this library is created. Use the OS/400 command:

```
WRKLIB QWAS6
```

Default profile creation

When WebSphere Application Server features are installed, certain default profiles are also created.

If the Core Product feature is installed, refer to the next section, “Core product profile creation”. If either the Web server plug-ins or Application Client features are installed, without the Core Product feature, refer to “HTTP and client profile creation” on page 267.

Core product profile creation

To ensure that the WebSphere Application Server default profile is created successfully, verify the creation of the profile in the following sections:

- For WebSphere Application Server Express or WebSphere Application Server (Base), see “Verifying the creation of the profile: Part 1”.
- For WebSphere Application Server Network Deployment, see “Verifying the creation of the profile: Part 2”.

Verifying the creation of the profile: Part 1

If either WebSphere Application Server Express or WebSphere Application Server (Base) is installed, a default profile is created during the installation.

Check the /QIBM/UserData/WebSphere/AppServer/V6/Base/profileRegistry/logs/wasprofile directory for the *wasprofile_create_default.log* log file. In this log file, search for INSTCONFSUCCESS. This parameter should be located at or near the end of log file. The entire message string is:

```
<message> INSTCONFSUCCESS: Success: The profile now exists.<message>
```

Verifying the creation of the profile: Part 2

If WebSphere Application Server Network Deployment is installed, two profiles are created during the installation. Check the /QIBM/UserData/WebSphere/AppServer/V6/ND/profileRegistry/logs/wasprofile directory for the following log files:

- wasprofile_create_default.log
- wasprofile_create_dmgr.log

In these log files, search for INSTCONFSUCCESS. The parameter should be at or near the end of the log file. The entire message string is:

```
<message> INSTCONFSUCCESS: Success: The profile now exists.<message>
```

After successful installation of the WebSphere Application Server V6 product, refer to “Installing fixes for WebSphere Application Server V6” on page 271.

HTTP and client profile creation

> **Note:** If the Core Product Files feature is installed, the HTTP and Client profiles are not created. These profiles are created only when the Web server plug-ins or Application Client feature is installed without the Core Product Files feature.

To ensure that the Web server plug-ins or Application Client default profile is created successfully, refer to one or both of the following sections:

- To verify profile creation for the Web server plug-ins feature, see “Verifying the creation of the HTTP profile”.
- To verify profile creation for the Application Client feature, see “Verifying the creation of the Client profile”.

Verifying the creation of the HTTP profile

If the Web server plug-ins feature is installed, without the Core Product feature, a default HTTP profile is created.

Check *one* of the following directories:

- /QIBM/UserData/WebSphere/AppServer/V6/Base/profileRegistry/logs/wasprofile
- /QIBM/UserData/WebSphere/AppServer/V6/ND/profileRegistry/logs/wasprofile

In the directory, look for the *wasprofile_create_http.log* log file. In this log file, search for INSTCONFSUCCESS. This parameter should be located at or near the end of the log file. The entire message string is:

```
<message> INSTCONFSUCCESS: Success: The profile now exists.<message>
```

Verifying the creation of the Client profile

If the Application Client feature is installed, without the Core Product Files feature, a client profile is created.

Check *one* of the following directories:

- /QIBM/UserData/WebSphere/AppServer/V6/Base/profileRegistry/logs/wasprofile
- /QIBM/UserData/WebSphere/AppServer/V6/ND/profileRegistry/logs/wasprofile

In the directory, look for the *wasprofile_create_client.log* log file. In this log file, search for INSTCONFSUCCESS. This parameter should be located at or near the end of the log file. The entire message string is:

```
<message> INSTCONFSUCCESS: Success: The profile now exists.<message>
```

After successful installation of the WebSphere Application Server V6 product, refer to “Installing fixes for WebSphere Application Server V6” on page 271.

WebSphere Application Server installation problems

The following sections identify problems that you might encounter when you install WebSphere Application Server V6.

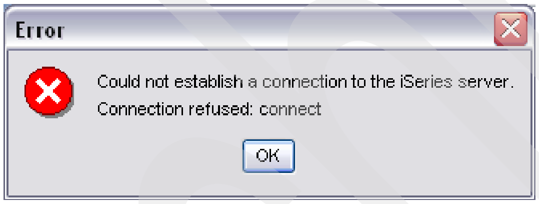
Host servers not started on iSeries

If the host servers are not started on the iSeries, the installation fails in one of the following ways:

- For a local iSeries installation, as described in “Command line installation from the iSeries” on page 264, you see the following message:

```
ServiceException: (error code = 3; message = "A remote host refused an attempted connect
operation."; severity = 0)
```

- For a command line installation, as described in “Command line installation from a remote workstation” on page 263, you see a window with an error message like the one in Figure B-11.
- For GUI installations, as described in “GUI installation from a remote workstation” on page 257, you see a window with an error message like the one in Figure B-11.



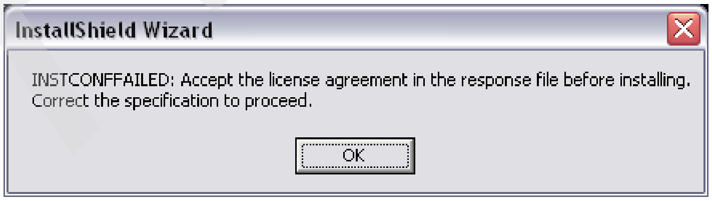
Could not establish a connection to the iSeries server.
Connection refused: connect

Figure B-11   Installation error message

License agreement not accepted

If the response file is not altered to accept the terms of the license agreement, the installation will fail in one of the following ways:

- For a silent command line installation, as described in “Command line installation from a remote workstation” on page 263
  - – You may see a window with an error message as shown in Figure B-12. In this case, edit the RESPONSEFILE. Refer to step 7 on page 263.



INSTCONFFAILED: Accept the license agreement in the response file before installing.
Correct the specification to proceed.

Figure B-12   License agreement error message

- If you do *not* see a message window like the one in Figure B-12 and you suspect that there is a problem, refer to the iSeries file /tmp/InstallShield/log.txt. The following message is logged:

  ```
  wrn, INSTCONFFAILED: Accept the license agreement in the response file before
  installing.
  ```

  Edit the response file. Refer to step 7 on page 263.

- For a local iSeries installation, as described in “Command line installation from the iSeries” on page 264, you see the following error message:

  ```
  INSTCONFFAILED: Accept the license agreement in the response file before installing.
  ```

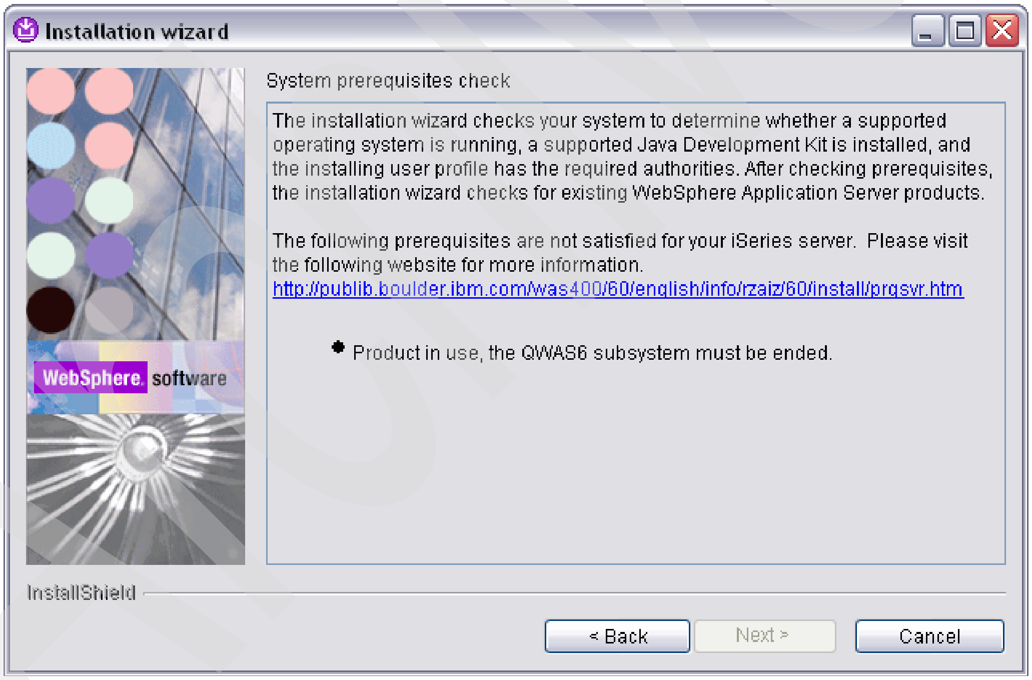
  Refer to step 6 on page 264 to alter the response file.

QWAS6 subsystem active

You must end the QWAS6 subsystem if additional features of WebSphere Application Server V6 are installed.

- For GUI installations, if the QWAS6 subsystem is active on the same iSeries system, you see a window like the one in Figure B-13.

  First end the jobs running in the QWAS6 subsystem, and then end the QWAS6 subsystem. Finally restart the installation.



System prerequisites check

The installation wizard checks your system to determine whether a supported operating system is running, a supported Java Development Kit is installed, and the installing user profile has the required authorities. After checking prerequisites, the installation wizard checks for existing WebSphere Application Server products.

The following prerequisites are not satisfied for your iSeries server. Please visit the following website for more information.
http://publib.boulder.ibm.com/was400/60/english/info/rzaiz/60/install/prqsvr.htm

- Product in use, the QWAS6 subsystem must be ended.

Figure B-13 QWAS6 subsystem active window

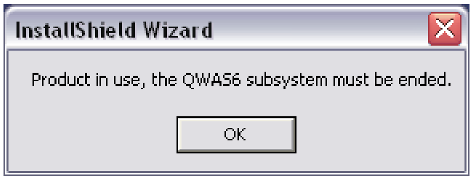
- For a local iSeries installation, as described in “Command line installation from the iSeries” on page 264, the installation fails with the following message:

  ```
  Product in use, the QWAS6 subsystem must be ended.
  ```

  First end the jobs in the QWAS6 subsystem and then end the QWAS6 subsystem. Finally, restart the installation.

- For a command line installation from a workstation as described in “Command line installation from a remote workstation” on page 263
  - You may see an error message like the one shown in Figure B-14.

    First end the jobs in the QWAS6 subsystem, and then end the QWAS6 subsystem. Finally, restart the installation.



InstallShield Wizard

Product in use, the QWAS6 subsystem must be ended.

Figure B-14 QWAS6 subsystem message

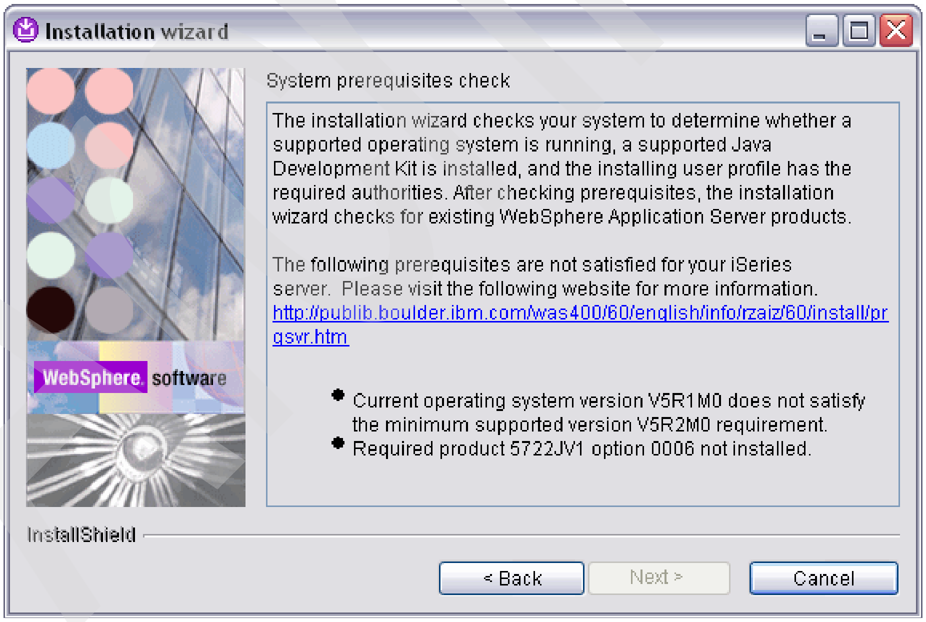
  - If you do not see an error message like the one shown in Figure B-14, and you suspect that there is a problem, look in the iSeries /tmp/InstallShield/log.txt file. You will see the that the following messages are logged:

```
err, SUBSYSTEM CHECK (support) : SubSystem QWAS6 found active.
err, Product in use, the QWAS6 subsystem must be ended.
```

Prerequisite check failures

If the proper prerequisites do not exist on the iSeries, the installation will fail in one of the following ways:

- For a GUI installation, you might see a panel like the one shown in Figure B-15. Correct the problems as listed in the window and attempt the installation again.



Installation wizard

System prerequisites check

The installation wizard checks your system to determine whether a supported operating system is running, a supported Java Development Kit is installed, and the installing user profile has the required authorities. After checking prerequisites, the installation wizard checks for existing WebSphere Application Server products.

The following prerequisites are not satisfied for your iSeries server. Please visit the following website for more information. http://publib.boulder.ibm.com/was400/60/english/info/rzaiz/60/install/prqsvr.htm

- Current operating system version V5R1M0 does not satisfy the minimum supported version V5R2M0 requirement.
- Required product 5722JV1 option 0006 not installed.

Figure B-15 Prerequisite failure panel

- For a command line installation from a remote workstation, as described in “Command line installation from a remote workstation” on page 263, check the /tmp/InstallShield/log.txt file on the iSeries if a prerequisite problem is suspected.

  In this log.txt file, you may see the message:

  ```
  Install, com.ibm.ws.install.ni.ismp.panels.OS400PrereqCheckPanel, err, S_PREREQ_FAIL
  ```

  To recover from this error, refer to “iSeries prerequisites” on page 256.

Installing fixes for WebSphere Application Server V6

After you install WebSphere Application Server V6 on your iSeries, it is important that you apply the latest Group PTF. Depending on the level of the OS/400 operating system, order one of the following Group PTF packages:

- SF99301 for V5R3M0
- SF99300 for V5R2M0

As in past versions of WebSphere Application Server, the Group PTF contains other Group PTFs. When you apply the Group PTF, you also apply WebSphere Application Server PTFs and PTFs for the HTTP Server, Java, and DB2 UDB products.

Note: when applying the WebSphere Application Server V6 Group PTF, you must follow additional instructions. For more details, see the Preventive Service Planning (PSP) information at:

http://www-912.ibm.com/s_dir/sline003.NSF/GroupPTFs?OpenView&Start=1&Count=30&Collapse=1#1

To install the WebSphere Application Server V6 Group PTF, complete the following steps:

1. Review the PSP information for the appropriate Group PTF SF99301 or SF99300.
2. Apply the Group PTF using normal PTF application methods.
3. Ensure that the group PTF is shown on the iSeries.

   ```
   WRKPTFGRP
   ```

   Look for either SF99300 (for V5R2M0) or SF99301 (for V5R3M0) and ensure that the status is *Installed*.
4. After you apply the Group PTF, the /QIBM/ProdData/WebSphere/AppServer/V6/*edition*/updateInstaller directory is created. In this directory, *edition* is one of the following types:
   - *Base* represents WebSphere Application Server Express or WebSphere Application Server (Base)
   - *ND* represents WebSphere Application Server Network Deployment.
5. End the QWAS6 subsystem if it is active.
6. Start a Qshell session on the iSeries.

7. Run the following command:

   ```
   cd /QIBM/ProdData/WebSphere/AppServer/edition/Base/updateInstaller
   ```

   Here *edition* is one of the following types:

   – *Base* represents WebSphere Application Server Express or WebSphere Application Server (Base)

   – *ND* represents WebSphere Application Server Network Deployment

8. Type the following command:

   ```
   update
   ```

   Important: Your system must meet the minimum hardware prerequisites. If it doesn't, the installation of the WebSphere fixes with the update script will take a *very* long time.

   The installation starts, and messages are displayed to indicate the progress. When the setup is complete, the last install message line is:

   ```
   Install, com.ibm.ws.install.ni.ismp.actions.ISMPLogSuccessMessageAction, msg1,
   INSTCONFSUCCESS
   ```

9. If there are existing WebSphere Application Server profiles on the system, further updates may be required. Refer again to the PSP information for additional instructions.

10. Exit the Qshell.

Checking the level of WebSphere Application Server V6

At times, you may need to know the level of WebSphere Application Server V6 that is installed on the iSeries. When the WebSphere Application Server V6 product is first installed on the iSeries system, its level is 6.0.0.1. If you have applied additional Group PTFs, refer to one or more of the following items to determine which level is installed:

- For WebSphere Application Server Core Product installed, see the "WAS.product file".
- For Web server plug-ins installed without Core Product, see the "PLG.product file".
- For Application Client installed without Core Product, see the "CLIENT.product file".
- For WebSphere edition, see the "versionInfo script".

Note: When a Group PTF has been applied, make sure that you complete the step to run the update script (run in Qshell). This update script updates the product files as explained in the following section. Applying only the PTFs from the Group PTF CDs does not update the product files.

WAS.product file

Check one or both of the following directories:

- /QIBM/UserData/WebSphere/AppServer/V6/Base/properties/version/WAS.product
- /QIBM/UserData/WebSphere/AppServer/V6/ND/properties/version/WAS.product

In the WAS.product file, search for the version line, for example:

```
<version>6.0.1.0</version>
```

PLG.product file

Check one or both of the following directories:

- /QIBM/UserData/WebSphere/AppServer/V6/Base/properties/version/PLG.product
- /QIBM/UserData/WebSphere/AppServer/V6/ND/properties/version/PLG.product

In the PLG.product file, search for the version line, for example:

```
<version>6.0.1.0</version>
```

CLIENT.product file

Check one or both of the following directories:

- /QIBM/UserData/WebSphere/AppServer/V6/Base/properties/version/CLIENT.product
- /QIBM/UserData/WebSphere/AppServer/V6/ND/properties/version/CLIENT.product

In the CLIENT.product file, search for the version line, for example:

```
<version>6.0.1.0</version>
```

versionInfo script

You can also use the versionInfo script to verify the version and fix level of your WebSphere installation. You can run the following two commands in Qshell:

```
cd /QIBM/ProdData/WebSphere/AppServer/V6/edition/bin
versionInfo
```

Note that *edition* refers to the edition of the WebSphere installed on the system, which is either Base or ND.

Uninstalling WebSphere Application Server V6

Unlike the many methods that are available for installing WebSphere Application Server, there is only one supported method for uninstalling the product. You can uninstall only the WebSphere Application Server product or uninstall the product and all user data (profiles).

To uninstall WebSphere Application Server, do the following steps:

1. Sign on to the iSeries with a user that has *SECOFR authority.
2. Start the host servers on the iSeries, using the command:

   ```
   STRHOSTSVR *ALL
   ```

3. Ensure that the QWAS6 subsystem has been stopped.
4. Start a Qshell session
5. Enter the following command:

   ```
   /QIBM/ProdData/WebSphere/AppServer/V6/edition/bin/uninstall
   ```

   Here *edition* is one of the following types:

   – *Base* represents WebSphere Application Server Express or WebSphere Application Server (Base)

   – *ND* represents WebSphere Application Server Network Deployment.

> **Note:** If a combination of WebSphere Application Server editions is installed on the iSeries, you may need to run the uninstall command two times, once from the Base directory and once from the ND directory. This depends on the options that need to remain installed on the iSeries.

When the uninstallation is complete and depending on the editions that were uninstalled, ensure that one or both of the following directories have been removed:

- /QIBM/ProdData/WebSphere/AppServer/V6/Base
- /QIBM/ProdData/WebSphere/AppServer/V6/ND

Several directories, which contain user data, will still exist. You can use this data for the following reasons:

- If the WebSphere Application Server product is installed at a later time, the user data can be reused.
- The user data can be exported to another system. Refer to the WebSphere Application Server Information Center for details about the export function:

  `http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp`

If the user-defined data is not needed, you can remove the following directories manually:

- /QIBM/UserData/WebSphere/AppServer/V6/Base
- /QIBM/UserData/WebSphere/AppServer/V6/ND

Uninstalling WebSphere Application Server and user data

To uninstall the product and user data, use the following steps:

1. Sign on to the iSeries with a user that has *SECOFR authority.
2. Start the host servers on the iSeries, using the command:

   ```
   STRHOSTSVR *ALL
   ```
3. Ensure that the QWAS6 subsystem has been stopped.
4. Start a Qshell session.
5. Enter the following command on one line:

   ```
   /QIBM/ProdData/WebSphere/AppServer/V6/edition/bin/uninstall -W
   deleteallprofiles.active="true"
   ```

   Here *edition* is one of the following types:

   – *Base* represents WebSphere Application Server Express or WebSphere Application Server V6 (Base)

   – *ND* represents WebSphere Application Server V6 Network Deployment.

> **Note:** If a combination of WebSphere Application Server editions is installed on the iSeries, you may need to run the uninstall command two times, once from the Base directory and once from the ND directory. This depends on the options that need to remain installed on the iSeries.

When the uninstallation is complete, depending on the editions that were uninstalled, ensure that one or both of the directories have been removed:

- /QIBM/ProdData/WebSphere/AppServer/V6/Base
- /QIBM/ProdData/WebSphere/AppServer/V6/ND

The user information will be removed from the following directories:

- /QIBM/UserData/WebSphere/AppServer/V6/Base/profiles
- /QIBM/UserData/WebSphere/AppServer/V6/ND/profiles

You can remove the directories manually from the iSeries.

C

Installing the Trade6 application

This appendix takes you step-by-step through the procedure to install the Trade6 sample application on a cluster of WebSphere Application Server profiles. To begin the steps in this appendix, you must first download and unzip the Trade6 files as explained in Appendix F, “Additional material” on page 297.

> **Note:** This appendix is based on the assumption that you’ve followed the example in the book and are ready to install this application after you have created the cluster.

Installing Trade6

After you successfully download and unzip the Trade6 files, you can install the Trade6 application to your WebSphere cluster.

Trade6 comes with several Java command language (JACL) scripts that make the installation easy. Follow these instructions to install Trade6 using the scripts:

1. Move the unzipped folder, *tradeinstall*, to the Integrated File System (IFS) of your Deployment Manager system. You can use any method, such as File Transfer Protocol (FTP) or network shared drive. For our example, we place tradeinstall into the /home directory.
2. When the folder is on the iSeries, open the installTrade60.sh file from the tradeinstall directory in any text editor. Modify the following line according to your deployment manager's name:

   ```
   export profile=dmgr
   ```

3. Save the file.
4. Sign on to the 5250 display of your deployment manager system and start Qshell:

   ```
   strqsh
   ```

5. Change to the tradeinstall directory.

   ```
   cd /home/tradeinstall
   ```

6. Run the following script:

   ```
   installTrade60.sh
   ```

7. The script connects to the deployment manager process and prompts for the first parameter:

   ```
   Global security is (or will be) enabled (true|false) [false]:
   ```

   Press Enter to accept the default value.
8. The next questions is:

   ```
   Is this a cluster installation (yes|no) [no]:
   ```

   Type yes and press Enter.
9. The next prompt is:

   ```
   Have all nodes been federated and network connectivity verified? (yes|no) [yes]:
   ```

   Press Enter.
10. You see the next prompt:

    ```
    Please enter the cluster name [TradeCluster]:
    ```

    If your cluster name is TradeCluster, then press Enter. Otherwise, type the cluster name and press Enter.
11. Provide some information about the cluster members. The script prompts:

    ```
    Current Cluster Nodes and Members:
         APPSERVER1_AppNode1 - Member1

    Add more cluster members (yes|no) [yes]:
    ```

Press Enter to add more members. You are prompted to add the next node name. In this example, we want to add Member2, which is on APPSERVER1_AppNode1.

```
Available Nodes:
   APPSERVER1_AppNode1
   APPSERVER2_AppNode2
   dmgrManager

Select the desired node [APPSERVER1_AppNode1]:
```

For the next member, Member2, press Enter.

When you add Member3 and Member4, you need to type a different node name, APPSERVER2_AppNode2.

12. The script prompts you for the name of the next cluster member:

    ```
    inside node member pairs...

     Please enter the cluster member name [TradeServer2]:
    ```

    Type the name of the next cluster member, Member2 in this case, and press Enter.

13. Repeat step 11 through step 12 for the other cluster members.
14. When all members are added, reply No to the following question:

    ```
    Add more cluster members (yes|no) [yes]:
    ```

15. Define the data source related information. The next prompt is:

    ```
    Select the backend database type
    (db2|oracle|db2cli|cloudscape|iSeriesNative|iSeriesToolbox) [db2]:
    ```

    Type `iSeriesToolbox` and press Enter. We use the Toolbox JDBC driver.

16. For the next prompt, press Enter:

    ```
    Please enter the database driver path [/QIBM/ProdData/HTTP/Public/jt400/lib/jt400.jar]:
    ```

    The implementation classes should be at the default location.

17. The next prompt is:

    ```
    Please enter the database schema [tradedb]:
    ```

    Type `TRADE` and press Enter.

18. The next prompt is a bit tricky:

    ```
    Please enter the Toolbox database hostname [localhost]:
    ```

    For our example, where we use cross-site mirroring (XSM), we need to provide the takeover IP address, not the host name of the primary independent auxiliary storage pool (IASP). We type `192.168.100.150`, and press Enter.

19. Provide authentication information for the next prompt:

    ```
    Please enter the database username [user]:
    ```

    Type a user name under which you access the application database and press Enter.

20. At the next prompt, type the password and press Enter.

    ```
    Please enter the database password [password]: .
    ```

At this point, the script starts running, creates several resources in WebSphere, and installs the application. When it is done, it restarts the deployment manager profile.

> **Attention:** This installation takes some time. It can be a long running action on the older hardware. Be patient and wait until the prompt comes back.

Restoring the database

One of the downloadable files that is included in the additional material for this book is a save file, trade51dbz.savf, that contains the Trade6 application database. To properly run the application, you should restore this database on the primary IASP. To download the file, see Appendix F, “Additional material” on page 297.

1. Create a save file on the iSeries where you’ve configured the primary IASP.
2. Send the trade51dbz.savf by FTP to your save file.
3. Restore the library by running the following command (replace highlighted parameters with you values):

   ```
   RSTLIB SAVLIB(TRADE51DBZ) DEV(*SAVF) SAVF(your lib/your save file) MBROPT(*ALL)
   ALWOBJDIF(*ALL) RSTASPDEV(TRADE)
   ```

The database should now be restored and ready to use.

Tip: For the first run of the application, start the **(Re)-populate Trade Database** configuration option from the application configuration page.

D

Installing Load Balancer

This appendix provides information and instructions for installing Load Balancer on a Windows workstation.

Preparing the environment

To start working with the product, you must verify several prerequisites, including hardware, software, and network settings.

Hardware and software

Table D-1 lists the hardware and software components that were used in the scenarios presented in this redbook.

Table D-1 Hardware and software components

Host	Role	Hardware	Software
EdgePrimary	Primary Load Balancer	NetVista Type 6792, 1 GB RAM, 40 GB HDD in two partitions; 6 GB and 32 GB, Ethernet 10/1000	► Windows 2000 server SP4, IE SP1, MS Security Fix, NAVv9x ► IBM WebSphere Edge Components v 6.0
EdgeBackup	BackUp Load Balancer	NetVista Type 6579, 1 GB RAM, 30 GB HDD in two partitions; 6 GB and 22 GB, Ethernet 10/1000	► Windows 2000 server SP4, IE SP1, MS Security Fix, NAVv9x ► IBM WebSphere Edge Components v 6.0
WebServer1	Web server (HTTP)		IBM WebSphere HTTP server v 6.0
WebServer2	Web server (HTTP)		IBM WebSphere HTTP server v 6.0

For more information about hardware and software prerequisites, see the following site:

http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html

Network

Configure the network adapters on four servers. In our case, we assume the network configuration that is listed in Table 5-3 on page 96.

We need to ensure that the following actions occur:

- Ensure that the host name of each server is correctly configured.
- Update the host table on each server to include the host names and IP addresses of other servers.
- Ensure that all servers can ping each other.

Installing Load Balancer

In this book, we explain how you can install Load Balancer on a Windows workstation. Before you begin the installation procedure, verify the following items:

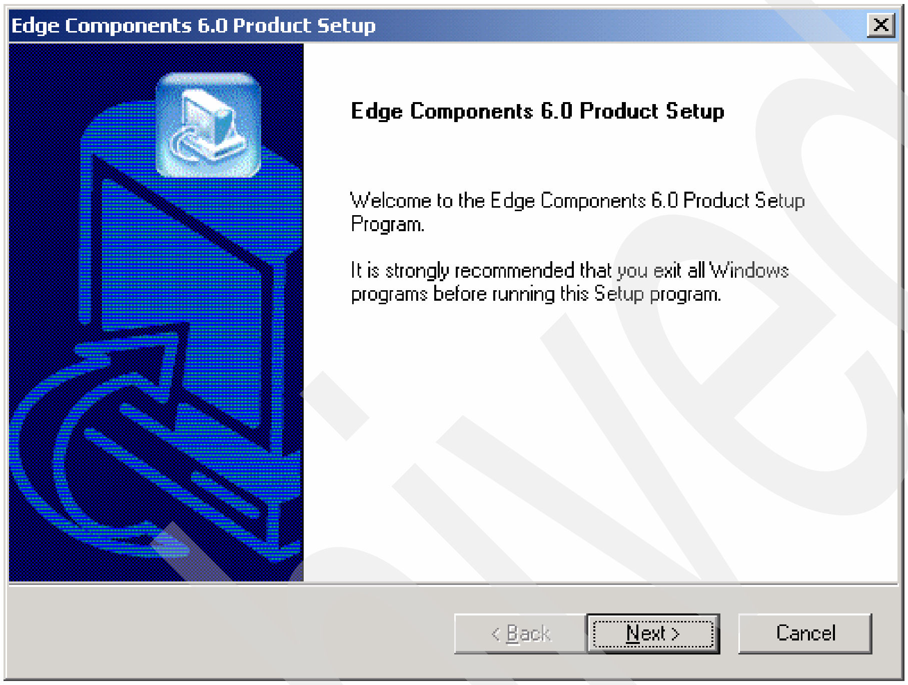
- IBM Firewall is not installed on the system. The Windows version of Load Balancer cannot be installed on the same machine with IBM Firewall.
- You can log in as the administrator or as a user with administrative privileges.
- If an earlier version of Load Balancer is installed, uninstall that copy before you install the current version.

Now, you need to perform the following installation steps:

1. Insert the WebSphere Application Server Edge Components CD into a CD-ROM drive on your workstation. If autorun is enabled, the wizard should start.

   Otherwise, go the root directory on your CD and double click **Setup.exe**. The Java 2 SDK automatically installs with Load Balancer on all platforms.

2. Edge Components Product Setup wizard starts. When you see the Welcome window (Figure D-1), click **Next**.


Edge Components 6.0 Product Setup

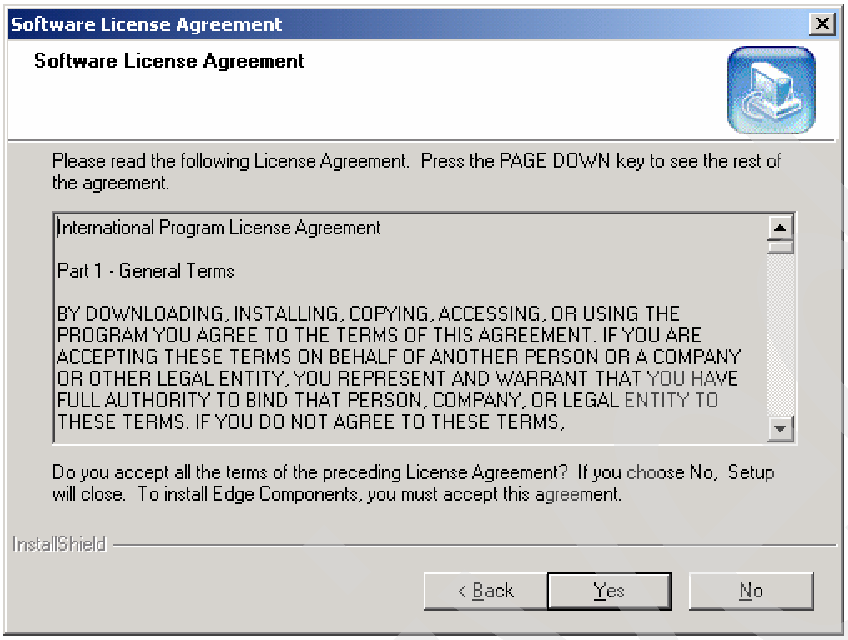
Edge Components 6.0 Product Setup

Welcome to the Edge Components 6.0 Product Setup Program.

It is strongly recommended that you exit all Windows programs before running this Setup program.

Figure D-1 Welcome window

3. In the Software License Agreement window (Figure D-2), read the agreement and click **Yes** to continue.



Software License Agreement

Software License Agreement

Please read the following License Agreement. Press the PAGE DOWN key to see the rest of the agreement.

International Program License Agreement

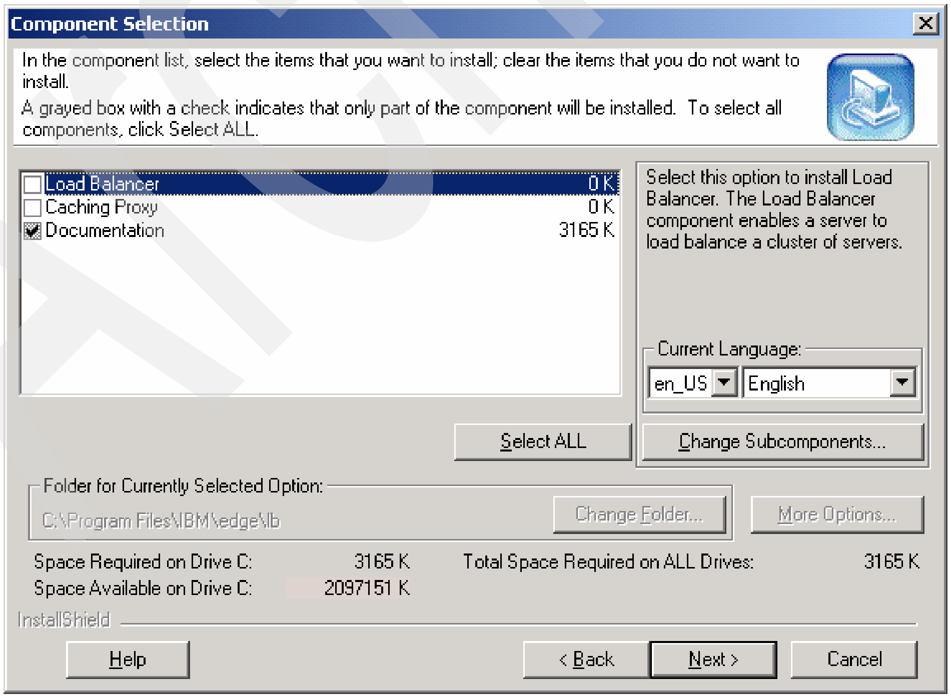
Part 1 - General Terms

BY DOWNLOADING, INSTALLING, COPYING, ACCESSING, OR USING THE PROGRAM YOU AGREE TO THE TERMS OF THIS AGREEMENT. IF YOU ARE ACCEPTING THESE TERMS ON BEHALF OF ANOTHER PERSON OR A COMPANY OR OTHER LEGAL ENTITY, YOU REPRESENT AND WARRANT THAT YOU HAVE FULL AUTHORITY TO BIND THAT PERSON, COMPANY, OR LEGAL ENTITY TO THESE TERMS. IF YOU DO NOT AGREE TO THESE TERMS,

Do you accept all the terms of the preceding License Agreement? If you choose No, Setup will close. To install Edge Components, you must accept this agreement.

Figure D-2 Software License Agreement window

4. In the Component Selection window (Figure D-3), you can select a component.
   a. Click the **Load Balancer** check box.
   b. Click **Change Subcomponents** to make changes in the Load Balancer subcomponents.



Component Selection

In the component list, select the items that you want to install; clear the items that you do not want to install.
A grayed box with a check indicates that only part of the component will be installed. To select all components, click Select ALL.

□ Load Balancer	0 K
□ Caching Proxy	0 K
☑ Documentation	3165 K

Select this option to install Load Balancer. The Load Balancer component enables a server to load balance a cluster of servers.

Current Language: en_US English

Folder for Currently Selected Option: C:\Program Files\IBM\edge\lb

Space Required on Drive C: 3165 K
Space Available on Drive C: 2097151 K
Total Space Required on ALL Drives: 3165 K

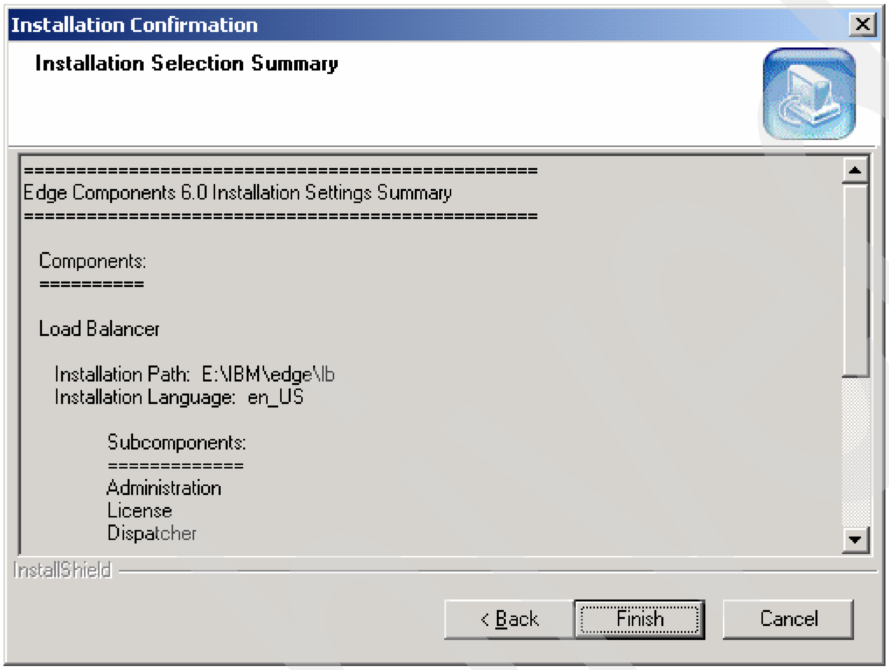
Figure D-3 Component Selection window

c. In the window that opens, for the EdgePrimary system, deselect the **Metric Server** subcomponent. The Metric Server is installed on the back-end servers to collect additional information. Click **OK**.

d. Back in the Component Selection window, verify that **Documentation** is selected.

e. Click **Next**.

5. The installation Confirmation window (Figure D-4) opens. Click **Finish**.



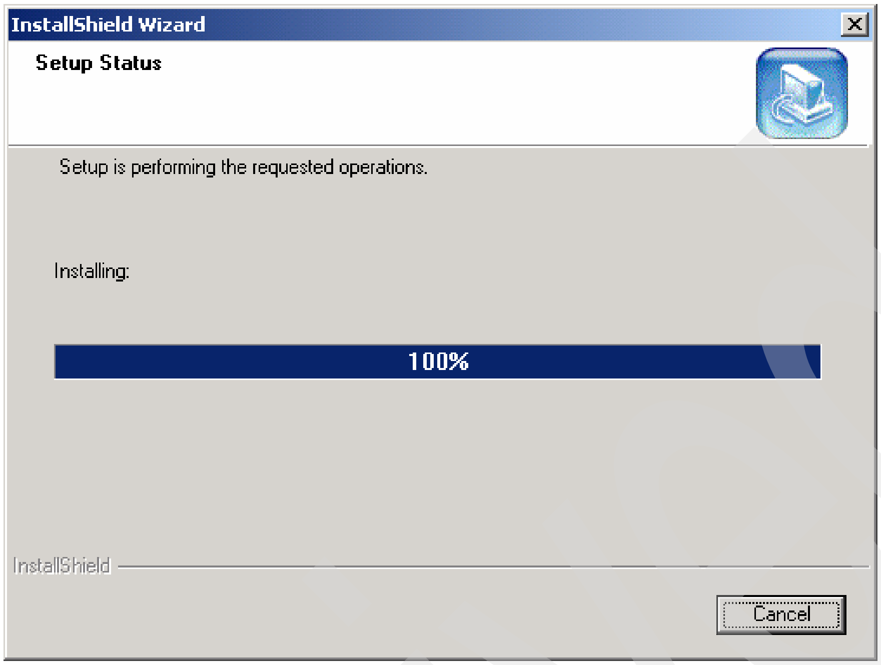
Installation Confirmation

Installation Selection Summary

```
=================================================
Edge Components 6.0 Installation Settings Summary
=================================================

  Components:
  ==========

  Load Balancer

    Installation Path:  E:\IBM\edge\lb
    Installation Language:  en_US

           Subcomponents:
           =============
           Administration
           License
           Dispatcher
```

Figure D-4 Installation Confirmation window

6. Wait for the installation process to end. Then the Setup Status window (Figure D-5) opens.



InstallShield Wizard

Setup Status

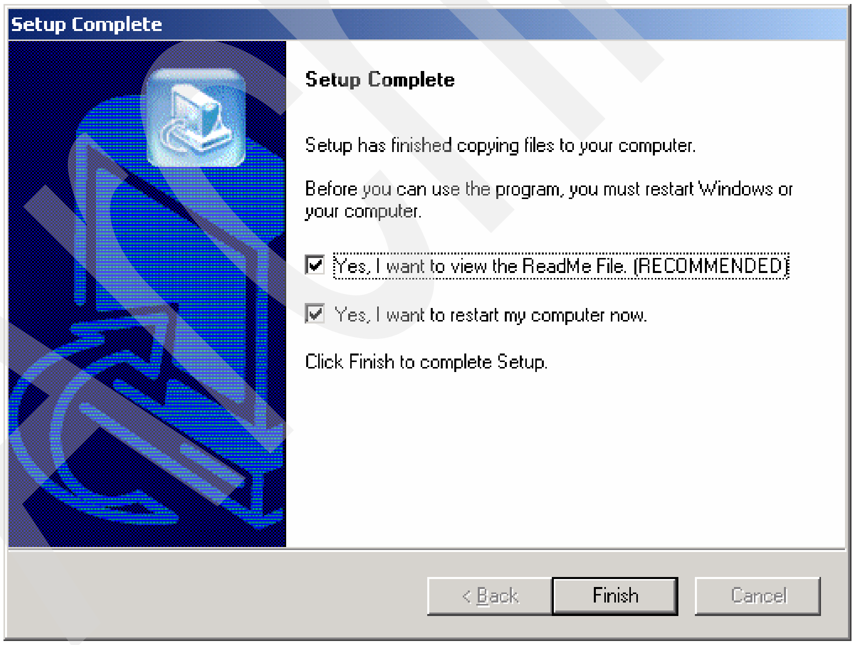
Setup is performing the requested operations.

Installing:

100%

Figure D-5 Setup Status window

7. When the installation process is finished, you should see the Setup Complete window (Figure D-6). Click **Finish**.



Setup Complete

Setup Complete

Setup has finished copying files to your computer.

Before you can use the program, you must restart Windows or your computer.

☑ Yes, I want to view the ReadMe File. (RECOMMENDED)

☑ Yes, I want to restart my computer now.

Click Finish to complete Setup.

Figure D-6 Setup Complete confirmation window

Review the readme file and then reboot the system. Then, your system is ready.

E

Additional geographic mirroring information

The network availability of your iSeries systems is an important component of achieving high availability. Technologies are available to help guarantee network routing capabilities even in the event of an iSeries network adapter failure. A virtual IP configuration, with proxy address resolution protocol (ARP), can help achieve that goal by having other network adapters in the same subnet take over requests for a failed network adapter. This way, end users are unaware of the network adapter failure.

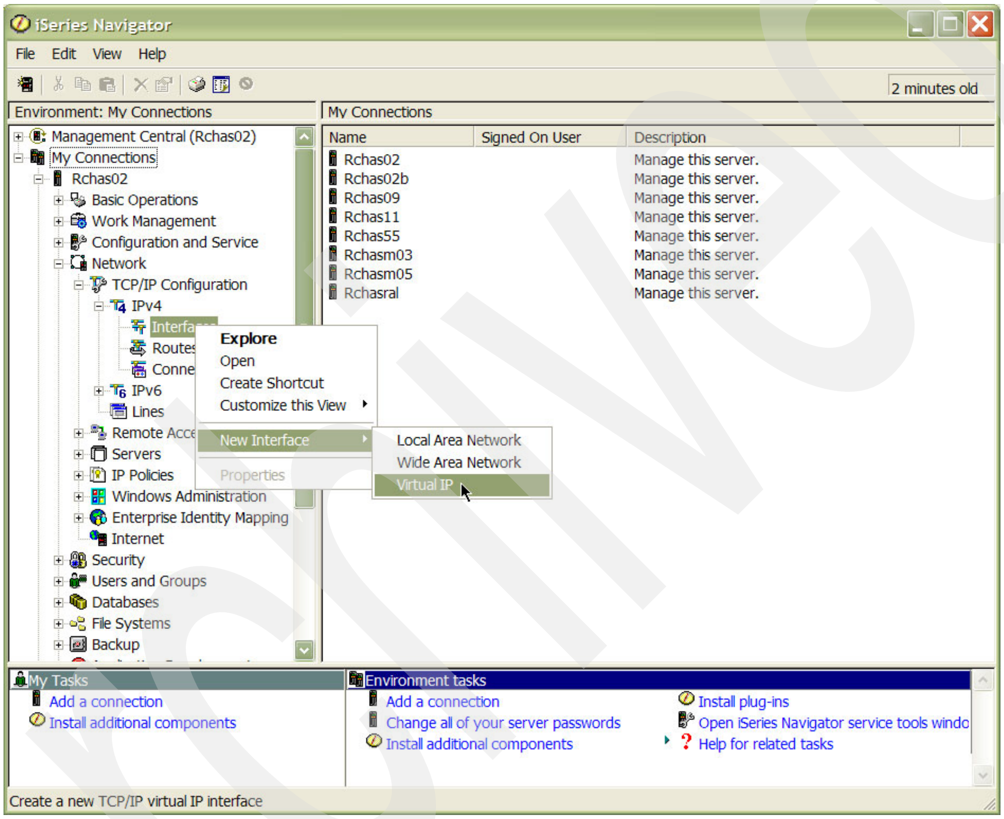
In addition to its application for regular communications transports, you can use this technology to increase the availability of the communications transports used for geographic mirroring replication processing between i5/OS cluster nodes. This appendix covers this as well as explains how to recover from an i5/OS cluster partition state, event after an unplanned system outage involving a two-node geographic mirroring configuration.

You can learn more about the configuration steps for i5/OS geographic mirroring in Chapter 5, “Scenario 2: Configuration for a large company” on page 93.

Configuring fault tolerance for network routes using virtual IP and proxy ARP

If you want a backup iSeries network adapter for each subnet, then you should configure a virtual IP with proxy ARP. We apply this technology to enable high availability of the communications transports used for cross-site mirroring (XSM) replication:

1. Configure a virtual TCP/IP interface with private address. In iSeries Navigator, as shown in Figure E-1, expand **My Connections** → ***your iSeries server*** → **Network** → **TCP/IP Configuration** → **IPv4**. Right-click **Interface** and choose **New Interface** → **Virtual IP**.



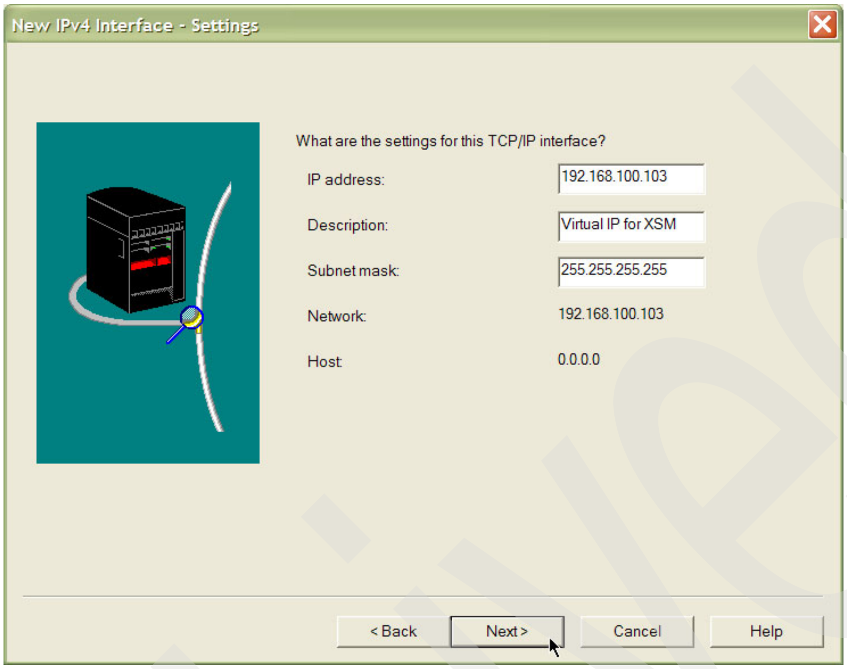
Name	Signed On User	Description
Rchas02		Manage this server.
Rchas02b		Manage this server.
Rchas09		Manage this server.
Rchas11		Manage this server.
Rchas55		Manage this server.
Rchasm03		Manage this server.
Rchasm05		Manage this server.
Rchasral		Manage this server.

Create a new TCP/IP virtual IP interface

Figure E-1 Adding a virtual IP interface

2. The wizard starts. Click **Next**.

3. In the New IPv4 Interface - Settings panel (Figure E-2), type the IP address and subnet mask for the new interface. The subnet mask must be 255.255.255.255 for the virtual IP interface. Use the values for your environment. Click **Next**.



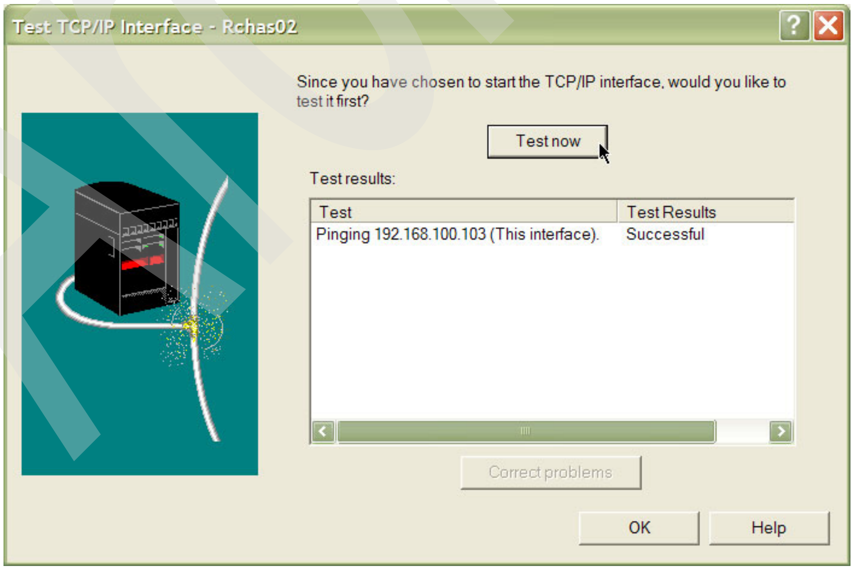
New IPv4 Interface - Settings

What are the settings for this TCP/IP interface?

IP address:	192.168.100.103
Description:	Virtual IP for XSM
Subnet mask:	255.255.255.255
Network:	192.168.100.103
Host:	0.0.0.0

Figure E-2   IP settings for the new interface

4. In the next panel, accept the default values and click **Next**.
5. In the Summary panel, review your settings and click **Finish**.
6. In the Test TCP/IP Interface panel (Figure E-3), click **Test now** to perform a test of the interface that we just configured. You see this panel only if you selected to start the interface in an earlier screen in the wizard. Click **OK** to end the wizard.



Test TCP/IP Interface - Rchas02

Since you have chosen to start the TCP/IP interface, would you like to test it first?

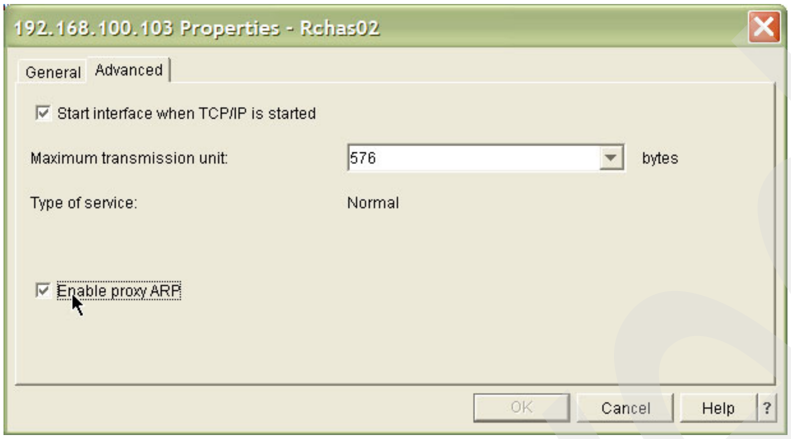
Test results:

Test	Test Results
Pinging 192.168.100.103 (This interface).	Successful

Figure E-3   Testing the interface

7. When the interface is created, you must to modify some of its properties. Stop the interface if it's active. Then right-click the virtual IP interface and select **Properties**.

8. In the Properties window (Figure E-4), click the **Advanced** tab. Select the **Enable proxy ARP** check box. Then click **OK**.



☑ Start interface when TCP/IP is started

Maximum transmission unit: 576 bytes

Type of service: Normal

☑ Enable proxy ARP

Figure E-4 Enabling the proxy ARP

9. Start your virtual IP interface.

Associating the local IP interfaces with the virtual IP interface

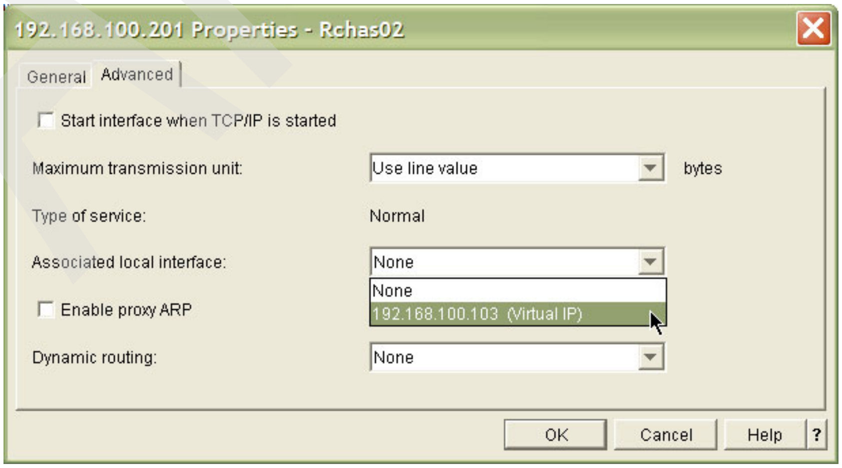
You can use up to four communications lines for XSM synchronization. To add a fault tolerance for these communication lines, you need to associate the same virtual IP address with all communications lines used for the synchronization. For more information, see *Routing with virtual IP* in the iSeries Information Center at:

`http://publib.boulder.ibm.com/infocenter/iseries/v5r3/index.jsp?topic=/rzajw/rzajwviproute.htm`

Perform these steps for each of your local IP interfaces that used for XSM synchronization:

1. Associate a local IP interface with the virtual IP interface that you just created. Stop your local IP interface. Then right-click your local IP interface and select **Properties**.

2. In the Properties window (Figure E-5), click the **Advanced** tab. For Associated local interface, select your virtual IP interface. Click **OK**.


192.168.100.201 Properties - Rchas02
General
Advanced
None
192.168.100.103 (Virtual IP)
OK
Cancel
Help
?

☐ Start interface when TCP/IP is started

Maximum transmission unit: Use line value bytes

Type of service: Normal

Associated local interface: None

☐ Enable proxy ARP

Dynamic routing: None

Figure E-5 Selecting the virtual IP interface

3. Start your local IP interface.

Now the requests that are sent to, in our example, 192.168.100.103, can be routed on any local IP interface that has the association with this virtual IP interface.

Updating your cluster configuration

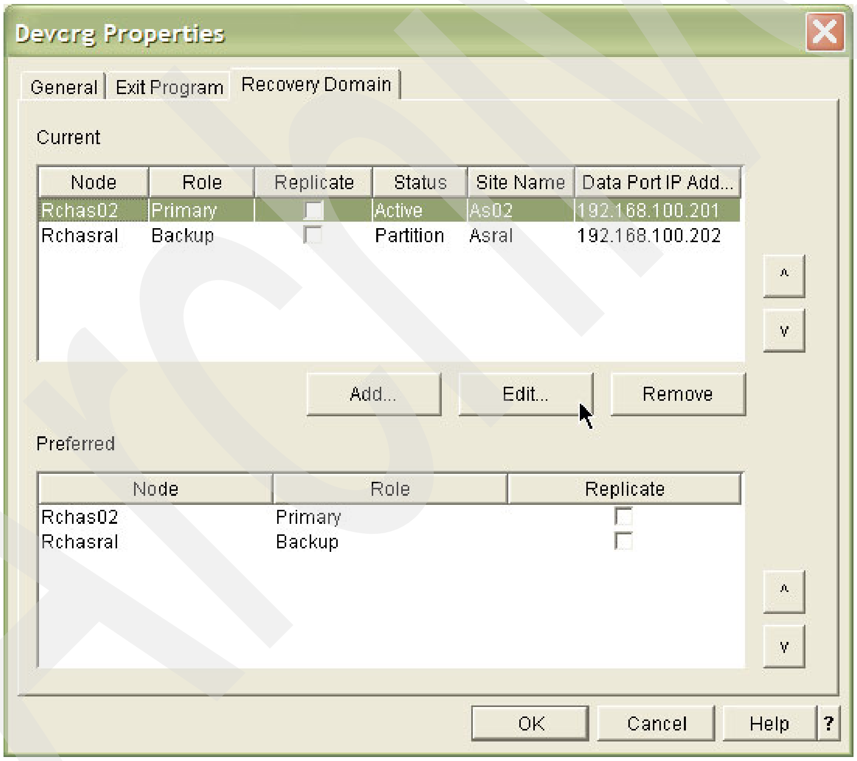
You need to modify the cluster settings to point to the virtual IP interface for data synchronization.

1. In iSeries Navigator, expand **Management Central** → **Clusters** → ***your cluster*** → **Switchable Hardware**.
2. Right-click your hardware group and select **Properties**.

> **Note:** If you perform these steps as part of the initial cluster configuration, then the properties window should be already opened.

3. In the Properties window (see Figure E-6), complete these steps:

   a. Select the **Recovery Domain** tab. Select the **Primary node** and click **Edit** (see Figure E-6).



Current

Node	Role	Replicate	Status	Site Name	Data Port IP Add...
Rchas02	Primary	☐	Active	As02	192.168.100.201
Rchasral	Backup	☐	Partition	Asral	192.168.100.202

Preferred

Node	Role	Replicate
Rchas02	Primary	☐
Rchasral	Backup	☐

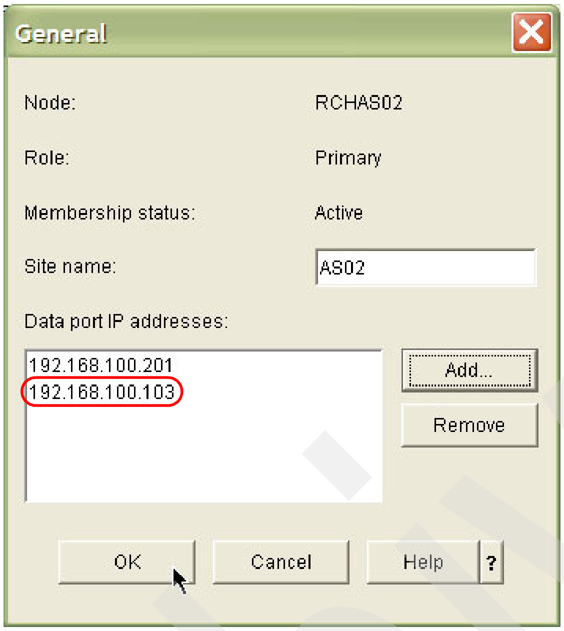
Figure E-6 Changing the cluster settings

b. In the General window (Figure E-7), follow these steps:

i. Click **Add** to add a new data port IP address.

ii. In the dialog box that opens, enter the virtual IP address. In this example, we type `192.168.100.103`. Then click **OK**.

iii. Now you should see all IP interfaces in the General window. Click **OK**.



General

Node: RCHAS02

Role: Primary

Membership status: Active

Site name: AS02

Data port IP addresses:

192.168.100.201

192.168.100.103

Figure E-7 Adding the virtual IP interface to the cluster configuration

c. Repeat steps a on page 291 through b on page 292 for the backup node.

d. In the Properties window, click **OK**.

> **Note:** At this point, if you already have XSM configured and running, you need to restart it to enable network routes using virtual IP and proxy ARP.

You have now completed a configuration that will provide higher availability for the communications transports used for XSM replication.

Using more than one communications line for synchronization

In the previous sections, you learned how to configure a fault tolerance for protecting one geographic mirroring synchronization line. You need to repeat these steps for each additional synchronization line. If you configure four communication lines for geographic mirroring (the maximum number in V5R3), you end up with four virtual IPs defined for the data ports on each cluster node.

The technical summary is that to add fault tolerance for each one of the communications lines used for geographic mirroring, you need to associate two local interfaces on different physical network adapters to a virtual IP address in the same subnet. Therefore, three interfaces are involved (behind the scenes) for *each* data port replication line:

- Two local interfaces on different physical adapters
- One virtual IP address, all in the same subnet

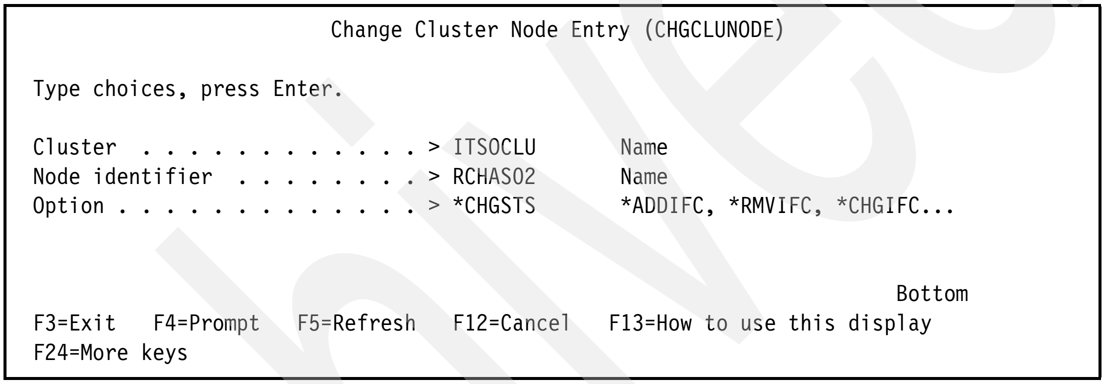
Recovery steps for a two-node geographic mirroring configuration

This information is related to a two-node geographic mirroring configuration when the primary cluster node goes into a partition state after an unplanned system outage. If the server that is hosting the primary cluster node fails due to a hard outage caused by a high-impact hardware or operating system failure, that node will likely go into *partition state*.

To recover from that failure and to enable users to access the independent disk pool again through the remaining cluster node, follow these steps:

1. Using the Change Cluster Node Entry (CHGCLUNODE) command (see Figure E-8) via the 5250 interface on the backup cluster node, change the status of the failed primary cluster node to *failed* from “partition”.

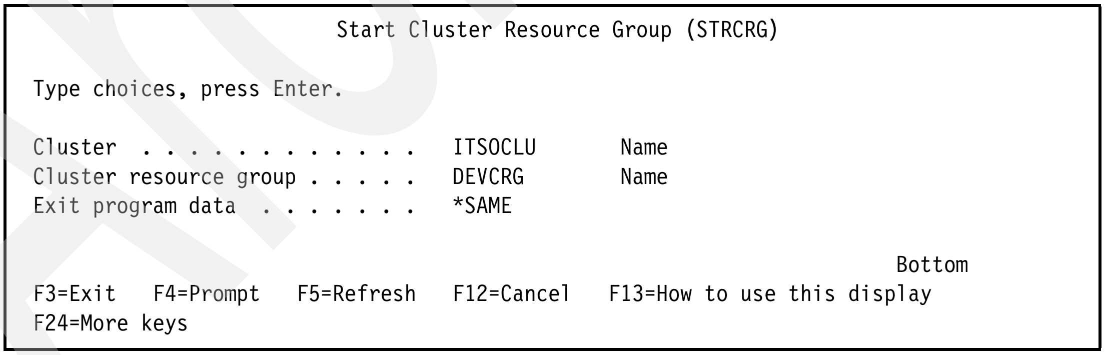
   On this panel, you must provide the cluster name, the name of the primary node, and the action that you want to perform (*CHGSTS).



```
                        Change Cluster Node Entry (CHGCLUNODE)

Type choices, press Enter.

Cluster  . . . . . . . . . . . . > ITSOCLU       Name
Node identifier  . . . . . . . . > RCHAS02       Name
Option . . . . . . . . . . . . . > *CHGSTS       *ADDIFC, *RMVIFC, *CHGIFC...



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

Figure E-8   Changing the cluster status

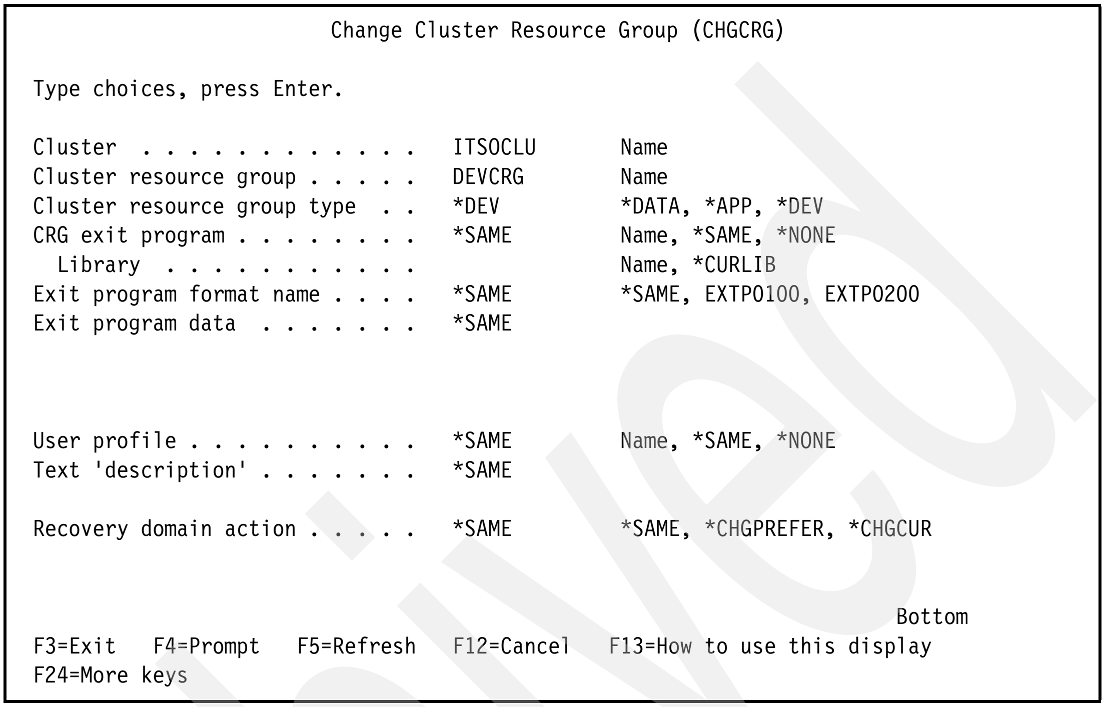
2. Start the device cluster resource group (CRG) on the backup node. Use the Start Cluster Resource Group (STRCRG) command (see Figure E-9).



```
                        Start Cluster Resource Group (STRCRG)

Type choices, press Enter.

Cluster  . . . . . . . . . . . .   ITSOCLU       Name
Cluster resource group . . . . .   DEVCRG        Name
Exit program data  . . . . . . .   *SAME

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

Figure E-9   Starting CRG

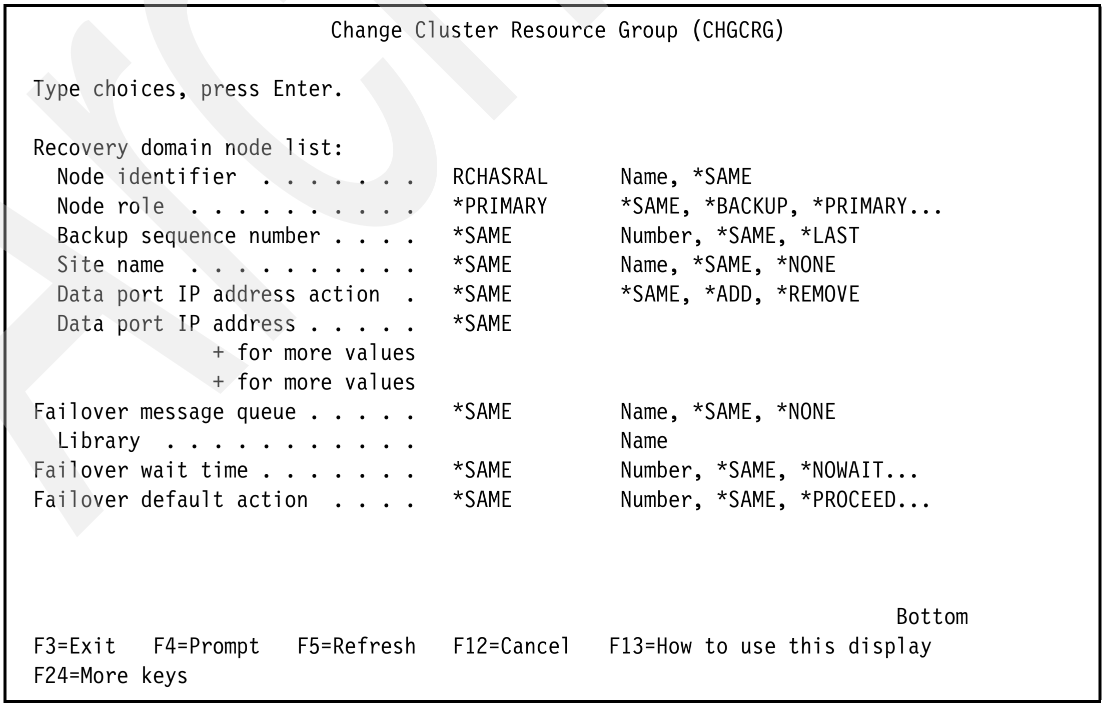
3. Switch the role of the backup node in the CRG to become the primary node. This can be done with the Change Cluster Resource Group (CHGCRG) command (see Figure E-10). Enter the Cluster name, CRG name, and the CRG type (*DEV). Then press Enter.



```
                          Change Cluster Resource Group (CHGCRG)

 Type choices, press Enter.

 Cluster  . . . . . . . . . . .   ITSOCLU       Name
 Cluster resource group . . . . .   DEVCRG        Name
 Cluster resource group type  . .   *DEV          *DATA, *APP, *DEV
 CRG exit program . . . . . . . .   *SAME         Name, *SAME, *NONE
   Library  . . . . . . . . . . .                 Name, *CURLIB
 Exit program format name . . . .   *SAME         *SAME, EXTP0100, EXTP0200
 Exit program data  . . . . . . .   *SAME



 User profile . . . . . . . . . .   *SAME         Name, *SAME, *NONE
 Text 'description' . . . . . . .   *SAME

 Recovery domain action . . . . .   *SAME         *SAME, *CHGPREFER, *CHGCUR


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

Figure E-10 Changing the CRG (part 1 of 2)

4. Page down for more options and change the Node Identifier to backup node name and Node Role to `*Primary` (see Figure E-11).



```
                          Change Cluster Resource Group (CHGCRG)

 Type choices, press Enter.

 Recovery domain node list:
   Node identifier  . . . . . . .   RCHASRAL      Name, *SAME
   Node role  . . . . . . . . . .   *PRIMARY      *SAME, *BACKUP, *PRIMARY...
   Backup sequence number . . . .   *SAME         Number, *SAME, *LAST
   Site name  . . . . . . . . . .   *SAME         Name, *SAME, *NONE
   Data port IP address action  .   *SAME         *SAME, *ADD, *REMOVE
   Data port IP address . . . . .   *SAME
                + for more values
                + for more values
 Failover message queue . . . . .   *SAME         Name, *SAME, *NONE
   Library  . . . . . . . . . . .                 Name
 Failover wait time . . . . . . .   *SAME         Number, *SAME, *NOWAIT...
 Failover default action  . . . .   *SAME         Number, *SAME, *PROCEED...




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

Figure E-11 Changing the CRG (part 2 of 2)

5. Having changed the role of the backup node to become the primary node, suspend geographic mirroring so that you can vary on the disk pool. Launch iSeries Navigator.

6. Expand **My Connections** → ***your backup iSeries*** → **Configuration and Service** → **Hardware** → **Disk Units** → **Disk Pools**. Use the service tools user profile to work with disk pools.

7. Right-click ***your IASP*** and select **Geographic Mirroring** → **Suspend Geographic Mirroring** (see Figure E-12).

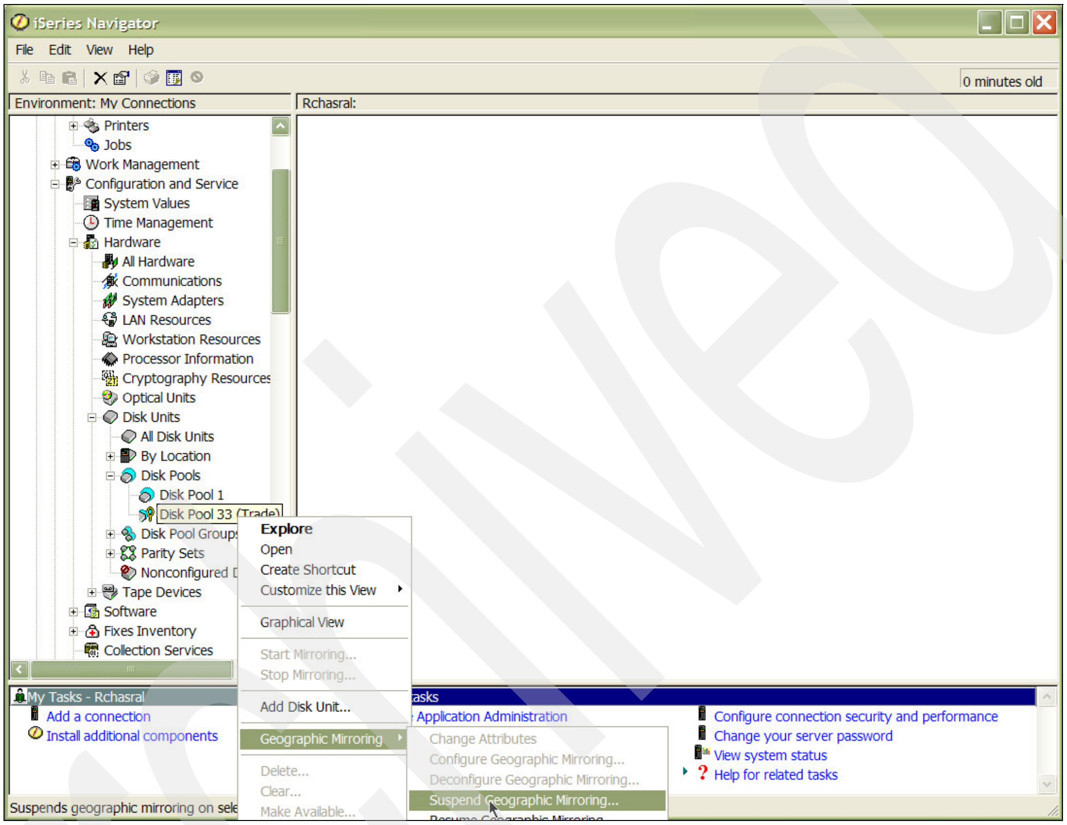
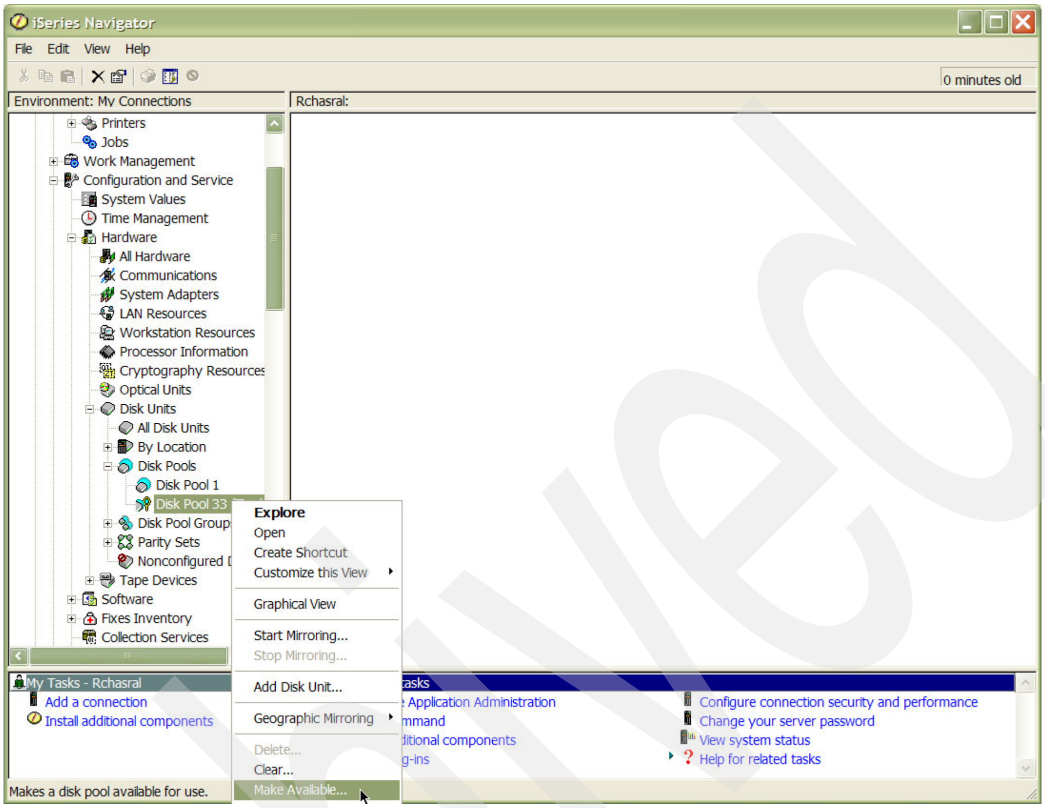


Figure E-12 Suspending geographic mirroring

8. Make the disk pool available. In iSeries Navigator, right-click the disk pool and select **Make Available** (see Figure E-13).



Makes a disk pool available for use.

Figure E-13 Varying on the disk pool

9. Make the database in the disk pool accessible to the WebSphere applications and other applications that refer to the takeover IP address. Manually start the IP takeover interface, by expanding **My Connections** → ***your backup iSeries*** → **Network** → **TCP/IP Configuration** → **IPv4**.

10. Click **Interfaces**.

11. When you see all configured interfaces in the right pane of the window, right-click the takeover IP interface and select **Start**.

At this point, your backup database is up and operational.

F

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/SG246637
```

Alternatively, you can go to the IBM Redbooks Web site at:

```
ibm.com/redbooks
```

Select the **Additional materials** and open the directory that corresponds with the redbook form number, SG246637.

Using the Web material

The additional Web material that accompanies this redbook includes the following files:

File name	***Description***
tradeinstall.zip	This file contains the installation scripts and the EAR file for the Trade6 application.
trade51dbz.savf	This is a save file, which contains the database used by the Trade6 application. See Appendix C, “Installing the Trade6 application” on page 277, for more information about restoring this database.

System requirements for downloading the Web material

The following system configuration is recommended:

Hard disk space:	30 MB of additional disk space for the downloaded files
Operating System:	Any of the latest Windows operating systems
Software:	Software to open a ZIP file

How to use the Web material

Create a subdirectory (folder) on your workstation, and unzip the contents of the tradeinstall.zip file into this folder. Continue with the instructions in Appendix C, “Installing the Trade6 application” on page 277.

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 about ordering these publications, see “How to get IBM Redbooks” on page 300. Note that some of the documents referenced here may be available in softcopy only.

- *Clustering and IASPs for Higher Availability on the IBM ⓔserver iSeries Server*, SG24-5194
- *WebSphere Application Server V6 Scalability and Performance Handbook,* SG24-6392
- *WebSphere Application Server V6: Planning and Design WebSphere Handbook,* SG24-6446
- *WebSphere Application Server V6 Network Deployment: High Availability Solutions,* SG24-6688
- *IBM HTTP Server (powered by Apache): An Integrated Solution for IBM ⓔserver iSeries Servers*, SG24-6716

Other publications

These publications are also relevant as further information sources:

- *Load Balancer Administration Guide*, GC31-6858
- *iSeries and High Availability: An e-Business Perspective*

  http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/product/iSeriesAndHa.pdf

Online resources

These Web sites and URLs are also relevant as further information sources:

- WebSphere Application Server for iSeries V6.0 documentation

  http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/docs/docws60.html
- IBM Patterns for e-business

  http://www.ibm.com/developerworks/patterns/index.html
- Dynamic logical partitioning

  http://www.ibm.com/servers/eserver/iseries/lpar/systemdesign.htm
- WebSphere Information Center

  http://www.ibm.com/software/webservers/appserv/doc/v60/ec/infocenter/index.html

- WebSphere Application Server for iSeries V6.0 system requirements

  http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html
- Rational Performance Tester site

  http://www.ibm.com/software/awdtools/tester/performance/
- iSeries Information Center

  http://publib.boulder.ibm.com/infocenter/iseries/v5r3/index.jsp
- OpenSTA

  http://www.opensta.org/
- Cisco Systems: Security appliances

  http://www.cisco.com/en/US/products/hw/vpndevc/ps2030/index.html
- Java Management Extensions (JMX)

  http://java.sun.com/products/JavaManagement/
- WebSphere Application Server document library

  http://www.ibm.com/software/webservers/appserv/was/library/
- iSeries: Preventative Service Planning - PSP

  http://www-912.ibm.com/s_dir/sline003.NSF/GroupPTFs?OpenView&Start=1&Count=30&C

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

IBM

Redbooks

WebSphere Application Server for iSeries V6: Building Advanced Configurations

(0.5" spine)
0.475"<->0.873"
250 <-> 459 pages

WebSphere Application Server for iSeries V6

Building Advanced Configurations

End-to-end setup instructions for complex configurations

Discussion of all phases in topology implementation

Cross-site mirroring configuration for high availability

Many IBM eserver iSeries clients have passed the initial phase of adoption of IBM WebSphere Application Server for iSeries. And now with *the* best practices presented in this book, they can build robust, high available solutions, based on WebSphere Application Server.

This IBM Redbook is designed to help system architects, WebSphere administrators, and software developers. It provides a detailed discussion about planning. Plus it provides implementation instructions to help build a complex solution, based on WebSphere Application Server for iSeries. In addition, it provides many useful techniques and tips for such an endeavor.

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

SG24-6637-00 ISBN 0738492973