WebSphere® software

IBM

Patterns for the Edge of Network

Runtime Patterns with WebSphere Edge Server

High availability guidelines

High performance guidelines

Richard Voegeli
Byron Braswell

Redbooks

ibm.com/redbooks

International Technical Support Organization

Patterns for the Edge of Network

November 2002

SG24-6822-00

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

First Edition (November 2002)

This edition applies to Version 2 of WebSphere Edge Server, Version 4 of WebSphere Application Server, and Version 3.9 of Access Manager.

Contents

Notices

Preface

Patterns for e-business are a group of proven, reusable assets that can help speed the process of developing applications.

In this IBM Redbook, we describe guidelines and options for the selection of Runtime patterns that include high availability and high performance considerations in the design process. Specifically, we address software and node configuration scenarios within the demilitarized zone (DMZ) and give examples of implementation procedures.

Part 1 of the redbook provides you with an overview of Patterns for e-business. It details Runtime patterns for high availability and high performance. It then gives you possible product mappings for implementation of the chosen runtime pattern.

Part 2 provides a set of guidelines for selecting your Runtime pattern within the DMZ. It includes information on technology options, high availability and high performance definitions, and a review of security considerations.

Part 3 takes you through installation and working examples, showing the implementation of high availability and high performance features of WebSphere Edge Server.

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, Raleigh Center.

Richard Voegeli is an Advisory IT Specialist in Switzerland. He has three years of experience in the e-business infrastructure field. He has worked at IBM for four years. His areas of expertise include WebSphere Application Server and WebSphere Edge Server on AIX, Solaris, Linux and Windows platforms.

Byron Braswell is a networking professional at the International Technical Support Organization, Raleigh Center. He received a B.S. degree in Physics and an M.S. degree in Computer Sciences from Texas A&M University. He writes extensively in the areas of networking and host integration software. Before joining the ITSO two years ago, Byron worked in IBM Learning Services Development in networking education development.

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

Margaret Ticknor
Michele Galic
Carla Sadtler
Mark Endrei
International Technical Support Organization, Raleigh Center

Jonathan Adams
IBM United Kingdom

Richard McDonald
IBM Canada

Kwabena Mireku
Chris Gage
Madhu Chetuparambil
Subbarao Meduri
Keith Smith
Kenichiroh Ueno
Gennaro Cuomo
John Connor
IBM Raleigh

Bill Hines
IBM Mechanicsburg

Become a published author

Comments welcome

Your comments are important to us!

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

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

  ibm.com/redbooks

- Send your comments in an Internet note to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HZ8 Building 662
  P.O. Box 12195
  Research Triangle Park, NC 27709-2195

Part 1

Introduction to Patterns

1

Patterns for e-business

This redbook is part of the *Patterns for e-business* series. In this introductory chapter, we provide an overview of how IT architects can work effectively with the Patterns for e-business.

The role of the IT architect is to evaluate business problems and build solutions to solve them. To do this, the architect begins by gathering input about the problem and creating an outline of the desired solution, taking into account any special considerations or requirements that need to be factored into that solution. The architect then takes this input and designs the solution. This solution can include one or more computer applications that address the business problems by supplying the necessary business functions.

To enable the architect to do this better each time, we need to capture and reuse the experience of these IT architects in such a way that future engagements can be made simpler and faster. We do this by taking these experiences and using them to build a repository of assets, a source from which architects can reuse this experience to build future solutions, using proven assets. This reuse saves time, money and effort and in the process helps ensure delivery of a solid, properly architected solution.

The IBM Patterns for e-business help facilitate this reuse of assets. Their purpose is to capture and publish e-business artifacts that have been used, tested and proven. The information captured by them is assumed to fit the majority, or 80/20 situation.

The IBM Patterns for e-business are further augmented with guidelines and related links for their better use.

The layers of patterns plus their associated links and guidelines allow the architect to start with a problem and a vision for the solution, and then find a pattern that fits that vision. Then, by drilling down using the patterns process, the architect can further define the additional functional pieces that the application will need to succeed. Finally, s/he can build the application using coding techniques outlined in the associated guidelines.

1.1 The Patterns for e-business layered asset model

The Patterns for e-business approach enables architects to implement successful e-business solutions through the re-use of components and solution elements from proven successful experiences. The Patterns approach is based on a set of layered assets that can be exploited by any existing development methodology. These layered assets are structured in a way that each level of detail builds on the last. These assets include:

- Business patterns that identify the interaction between users, businesses, and data.
- Integration patterns that tie multiple Business patterns together when a solution cannot be provided based on a single Business pattern.
- Composite patterns that represent commonly occurring combinations of Business patterns and Integration patterns.
- Application patterns that provide a conceptual layout describing how the application components and data within a Business pattern or Integration pattern interact.
- Runtime patterns that define the logical middleware structure supporting an Application pattern. Runtime patterns depict the major middleware nodes, their roles, and the interfaces between these nodes.
- Product mappings that identify proven and tested software implementations for each Runtime pattern.
- Best-practice guidelines for design, development, deployment, and management of e-business applications.

These assets and their relation to each other are shown in Figure 1-1.

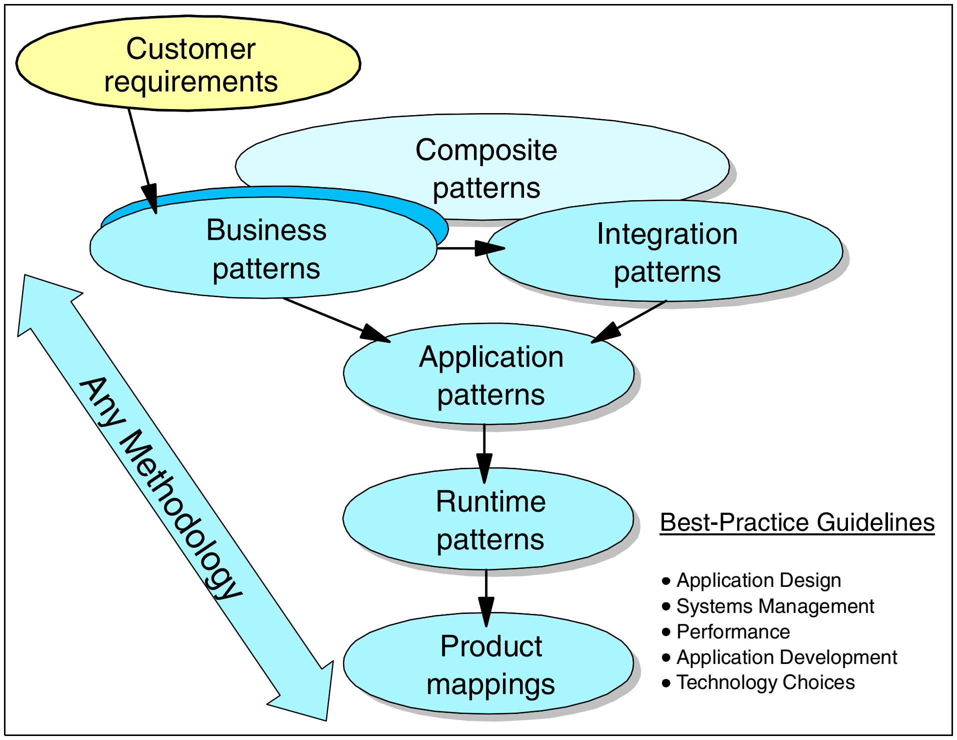


Figure 1-1 The Patterns for e-business layered asset model

Patterns for e-business Web site

The Patterns Web site provides an easy way of navigating top down through the layered Patterns' assets in order to determine the preferred reusable assets for an engagement.

For easy reference to Patterns for e-business, refer to the Patterns for e-business Web site at:

`http://www.ibm.com/developerWorks/patterns/`

1.2 How to use the Patterns for e-business

As described in the last section, the Patterns for e-business are a layered structure where each layer builds detail on the last. At the highest layer are Business patterns. These describe the entities involved in the e-business solution.

Composite patterns appear in the hierarchy shown in Figure 1-1 above the Business patterns. However, Composite patterns are made up of a number of individual Business patterns, and at least one Integration pattern. In this section, we will discuss how to use the layered structure of Patterns for e-business assets.

1.2.1 Select a Business, Integration, or Composite pattern, or a Custom design

When faced with the challenge of designing a solution for a business problem, the first step is to take a high-level view of the goals you are trying to achieve. A proposed business scenario should be described and each element should be matched to an appropriate IBM Pattern for e-business. You may find, for example, that the total solution requires multiple Business and Integration patterns, or that it fits into a Composite pattern or Custom design.

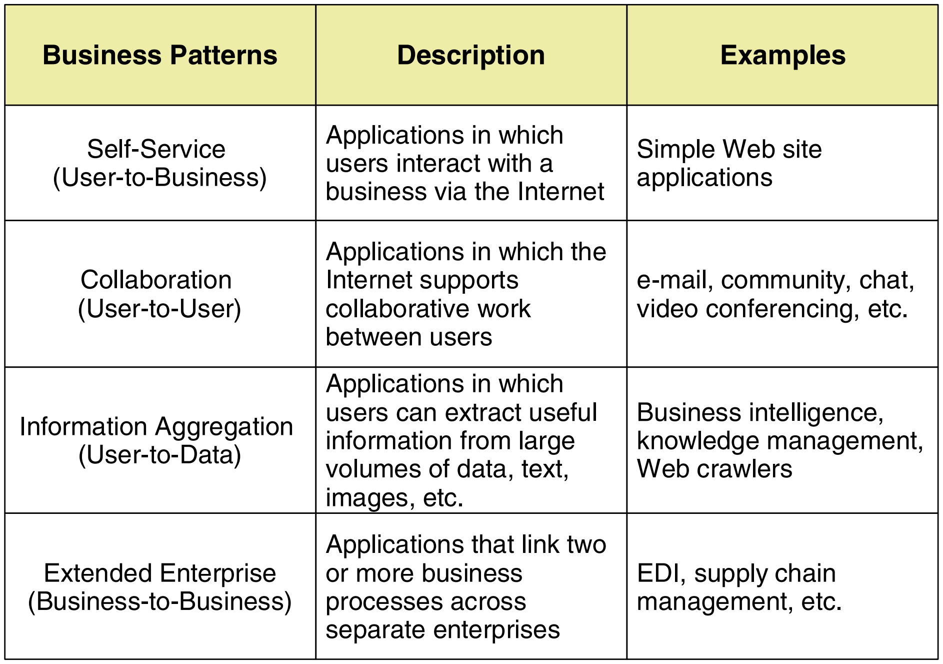
For example, suppose an insurance company wants to reduce the amount of time and money spent on call centers that handle customer inquiries. By allowing customers to view their policy information and to request changes online, they will be able to cut back significantly on the resources spent handling this by phone. The objective is to allow policyholders to view their policy information stored in legacy databases.

The Self-Service business pattern fits this scenario perfectly. It is meant to be used in situations where users need direct access to business applications and data. Let’s take a look at the available Business patterns.

Business patterns

A Business pattern describes the relationship between the users, the business organization or applications, and the data to be accessed.

There are four primary Business patterns.



Business Patterns	Description	Examples
Self-Service (User-to-Business)	Applications in which users interact with a business via the Internet	Simple Web site applications
Collaboration (User-to-User)	Applications in which the Internet supports collaborative work between users	e-mail, community, chat, video conferencing, etc.
Information Aggregation (User-to-Data)	Applications in which users can extract useful information from large volumes of data, text, images, etc.	Business intelligence, knowledge management, Web crawlers
Extended Enterprise (Business-to-Business)	Applications that link two or more business processes across separate enterprises	EDI, supply chain management, etc.

Figure 1-2 The four primary Business patterns

It would be very convenient if all problems fit nicely into these four slots, but in reality, things will often be more complicated. The patterns assume that most problems, when broken down into their most basic components, will fit more than one of these patterns. When a problem requires multiple Business patterns, the Patterns for e-business provide additional patterns in the form of Integration patterns.

Integration patterns

Integration patterns allow us to tie together multiple Business patterns to solve a business problem. The Integration patterns include:

Integration Patterns	Description	Examples
Access Integration	Integration of a number of services through a common entry point	Portals
Application Integration	Integration of multiple applications and data sources without the user directly invoking them	Message brokers, workflow managers

Figure 1-3 Integration patterns

These Business and Integration patterns can be combined to implement installation-specific business solutions. We call this a Custom design.

Custom design

We can represent the use of a Custom design to address a business problem through an iconic representation, as shown below:

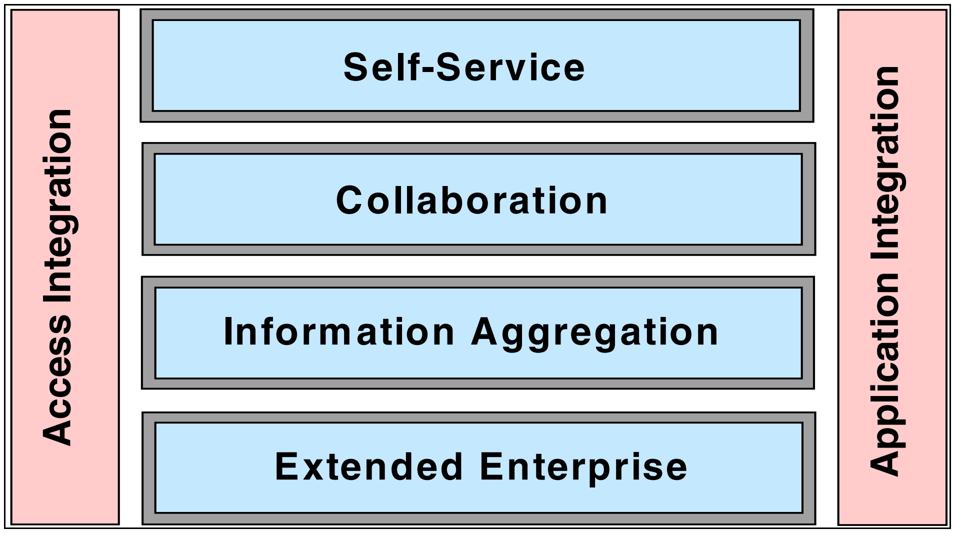


Figure 1-4 Patterns representing a Custom design

If any of the Business or Integration patterns are not used in a Custom design, we can show this with blocks that are lighter than the others. For example, Figure 1-5 shows a Custom design that does not have a Collaboration business pattern or an Extended Enterprise business pattern for a business problem:

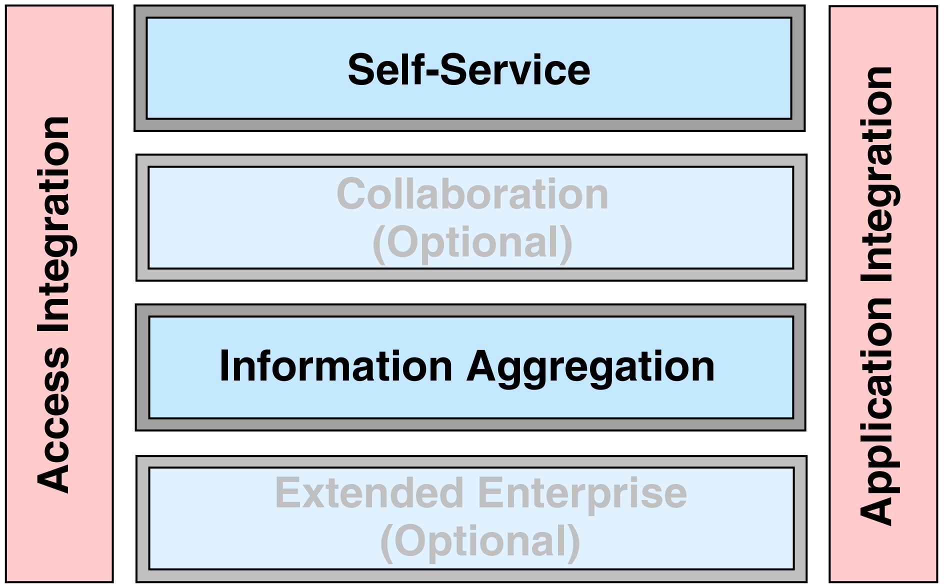
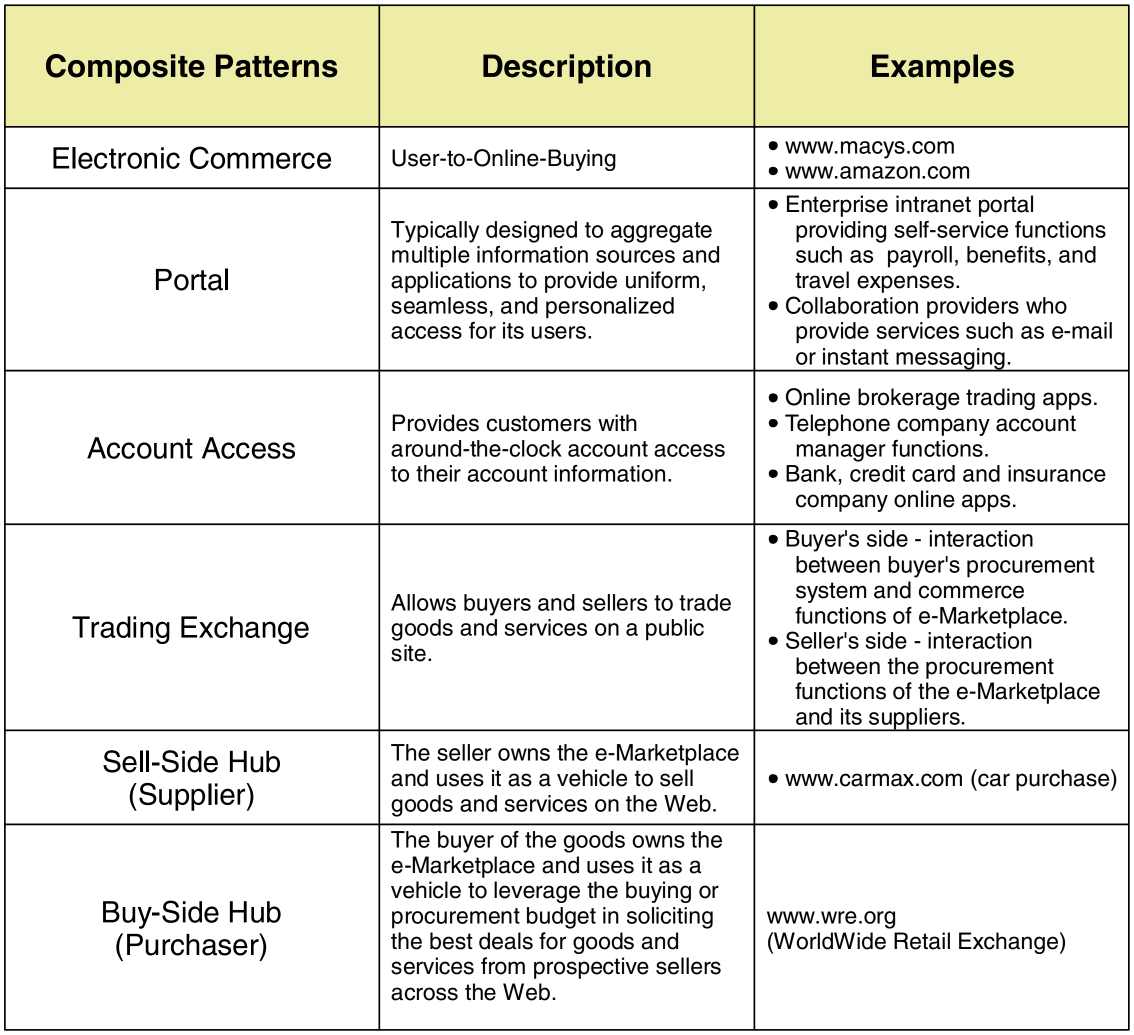


Figure 1-5 Custom design excluding Collaboration and Extended Enterprise patterns

A Custom design may also be a Composite pattern if it recurs many times across domains with similar business problems. For example, the iconic view of a Custom design in Figure 1-5 can also describe a Sell-Side Hub composite pattern.

Composite patterns

Several common uses of Business and Integration patterns have been identified and formalized into Composite patterns. The identified Composite patterns are as follows:



Composite Patterns	Description	Examples
Electronic Commerce	User-to-Online-Buying	• www.macys.com • www.amazon.com
Portal	Typically designed to aggregate multiple information sources and applications to provide uniform, seamless, and personalized access for its users.	• Enterprise intranet portal providing self-service functions such as payroll, benefits, and travel expenses. • Collaboration providers who provide services such as e-mail or instant messaging.
Account Access	Provides customers with around-the-clock account access to their account information.	• Online brokerage trading apps. • Telephone company account manager functions. • Bank, credit card and insurance company online apps.
Trading Exchange	Allows buyers and sellers to trade goods and services on a public site.	• Buyer's side - interaction between buyer's procurement system and commerce functions of e-Marketplace. • Seller's side - interaction between the procurement functions of the e-Marketplace and its suppliers.
Sell-Side Hub (Supplier)	The seller owns the e-Marketplace and uses it as a vehicle to sell goods and services on the Web.	• www.carmax.com (car purchase)
Buy-Side Hub (Purchaser)	The buyer of the goods owns the e-Marketplace and uses it as a vehicle to leverage the buying or procurement budget in soliciting the best deals for goods and services from prospective sellers across the Web.	www.wre.org (WorldWide Retail Exchange)

Figure 1-6 Composite patterns

The makeup of these patterns is variable in that there will be basic patterns present for each type, but the Composite can easily be extended to meet additional criteria. For more information on Composite patterns, refer to *Patterns for e-business: A Strategy for Reuse* by Jonathan Adams, Srinivas Koushik, Guru Vasudeva, George Galambos, ISBN: 1-931182-02-7.

1.2.2 Selecting Application patterns

Once the Business pattern is identified, the next step is to define the high-level logical components that make up the solution and how these components interact. This is known as the Application pattern. A Business pattern will usually have multiple possible Application patterns. An Application pattern may have logical components that describe a presentation tier for interacting with users, an application tier, and a back-end application tier.

Application patterns break the application down into the most basic conceptual components, identifying the goal of the application. In our example, the application falls into the Self Service business pattern and the goal is to build a simple application that allows users to access back-end information. The following Application pattern fulfills this requirement.

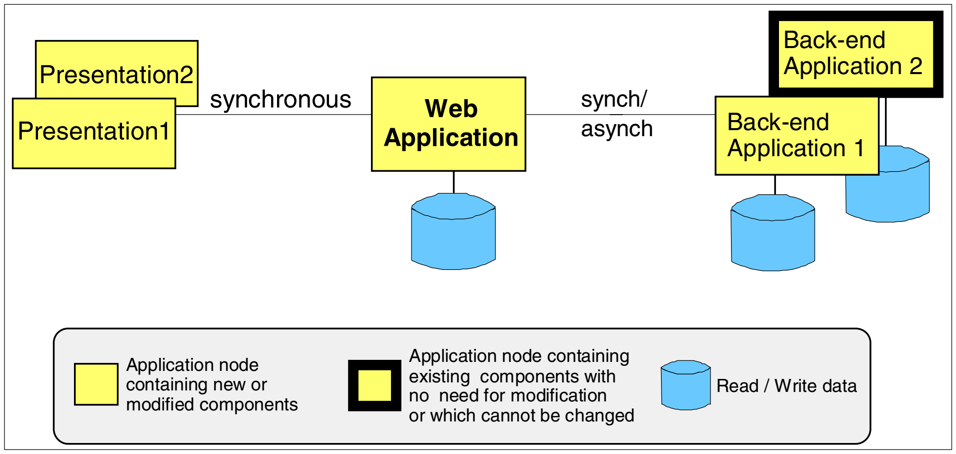

Presentation2
Presentation1
synchronous
Web Application
synch/ asynch
Back-end Application 2
Back-end Application 1
Application node containing new or modified components
Application node containing existing components with no need for modification or which cannot be changed
Read / Write data

Figure 1-7 Self -Service: directly integrated single channel

The Application pattern shown consists of a presentation tier that handles the request/response to the user. The application tier represents the component that handles access to the back-end applications and data. The multiple application boxes on the right represent the back-end applications that contain the business data. The type of communication is specified as synchronous (one request/one response, then the next request/response) or asynchronous (multiple requests and responses intermixed).

Suppose that the situation is a little more complicated than that. Let's say that the automobile policies and the homeowner policies are kept in two separate and dissimilar databases. The user request would actually need data from multiple, disparate back-end systems. In this case, there is a need to break the request down into multiple requests (decompose the request) to be sent to the two different back-end databases, then to gather the information sent back from the requests and put this information into the form of a response (recompose). In this case, the following Application pattern would be more appropriate.

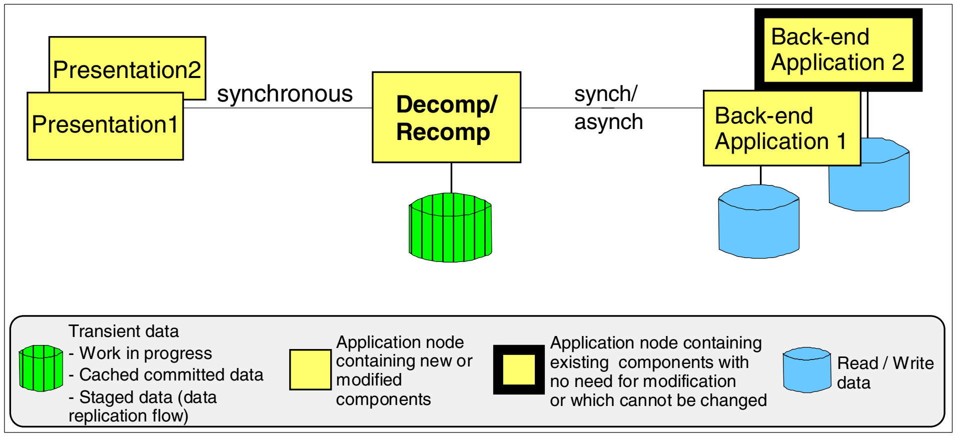

Presentation2
Presentation1
synchronous
Decomp/ Recomp
synch/ asynch
Back-end Application 2
Back-end Application 1
Transient data
- Work in progress
- Cached committed data
- Staged data (data replication flow)
Application node containing new or modified components
Application node containing existing components with no need for modification or which cannot be changed
Read / Write data

Figure 1-8 Self-Service: decomposition

This Application pattern extends the idea of the application tier that accesses the back-end data by adding decomposition and recomposition capabilities.

1.2.3 Review Runtime patterns

The Application pattern can be further refined with more explicit functions to be performed. Each function is associated with a runtime node. In reality, these functions, or nodes, can exist on separate physical machines or may co-exist on the same machine. In the Runtime pattern, this is not relevant. The focus is on the logical nodes required and their placement in the overall network structure.

As an example, let's assume that our customer has determined that his solution fits into the Self Service business pattern and that the Directly Integrated Single Channel pattern is the most descriptive of the situation. The next step is to determine the Runtime pattern that is most appropriate for his situation.

He knows that he will have users on the Internet accessing his business data and he will therefore require a measure of security. Security can be implemented at various layers of the application, but the first line of defense is almost always one or more firewalls that define who and what can cross the physical network boundaries into his company network.

He also needs to determine the functional nodes required to implement the application and security measures. The following Runtime pattern is one of his options.

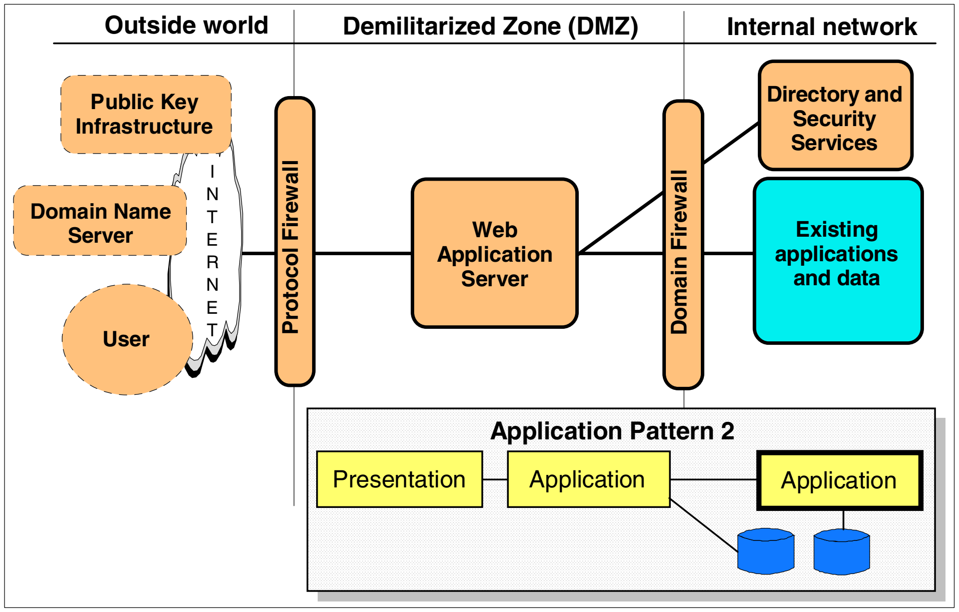


Figure 1-9 Directly Integrated Single Channel application pattern: Runtime pattern

By overlaying the Application pattern on the Runtime pattern, you can see the roles that each functional node will fulfill in the application. The presentation and application tiers will be implemented with a Web Application Server, which combines the functions of an HTTP server and an application server. It handles both static and dynamic Web pages.

Application security is handled by the Web Application Server through the use of a common central directory and security services node.

A characteristic that makes this Runtime pattern different from others is the placement of the Web Application Server between the two firewalls. The following Runtime pattern is a variation on this. It splits the Web Application Server into two functional nodes by separating the HTTP server function from the application server. The HTTP server (Web server redirector) will serve static Web pages and redirect other requests to the application server. It moves the application server function behind the second firewall, adding further security.

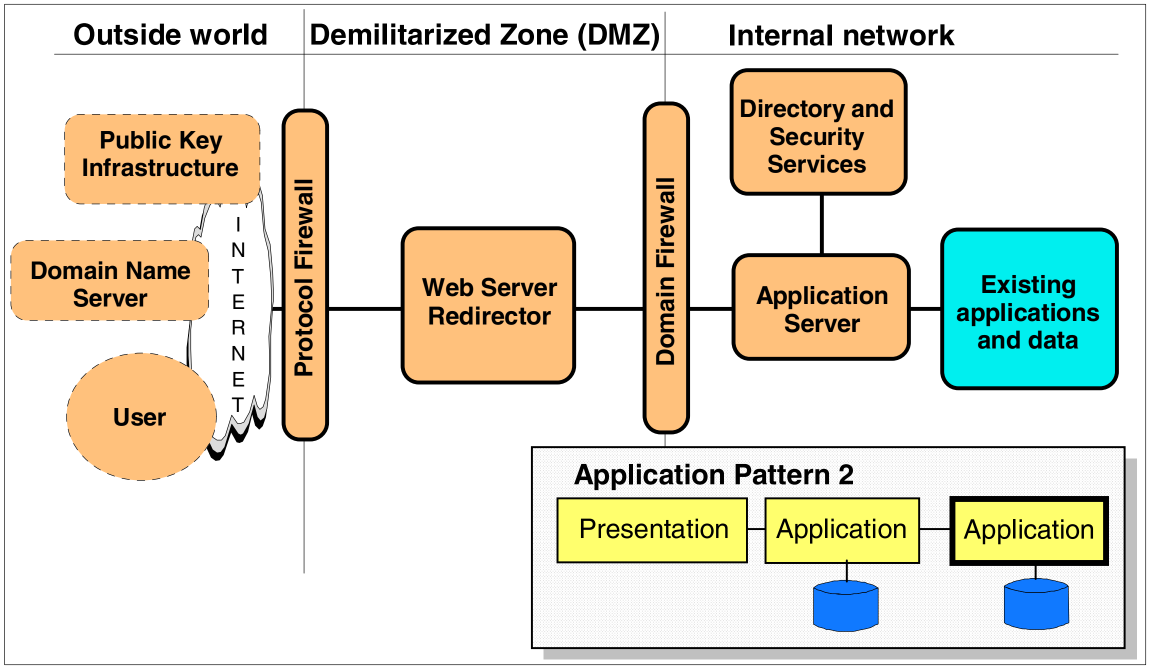


Figure 1-10 Directly Integrated Single Channel application pattern: Runtime Variation 2

These are just two examples of the possible Runtime patterns available. Each Application pattern will have one or more Runtime patterns defined. These can be modified to suit the customer needs. For example, he/she may want to add a load-balancing function and multiple application servers.

1.2.4 Review Product mappings

The last step in defining the network structure for the application is to correlate real products with one or more runtime nodes. The patterns Web site shows each Runtime pattern with products that have been tested in that capacity. The Product mappings are oriented toward a particular platform, though it is more likely that the customer will have a variety of platforms involved in the network. In this case, it is simply a matter of mix and match.

For example, the runtime variation in Figure 1-10 could be implemented using the product set depicted in Figure 1-11.

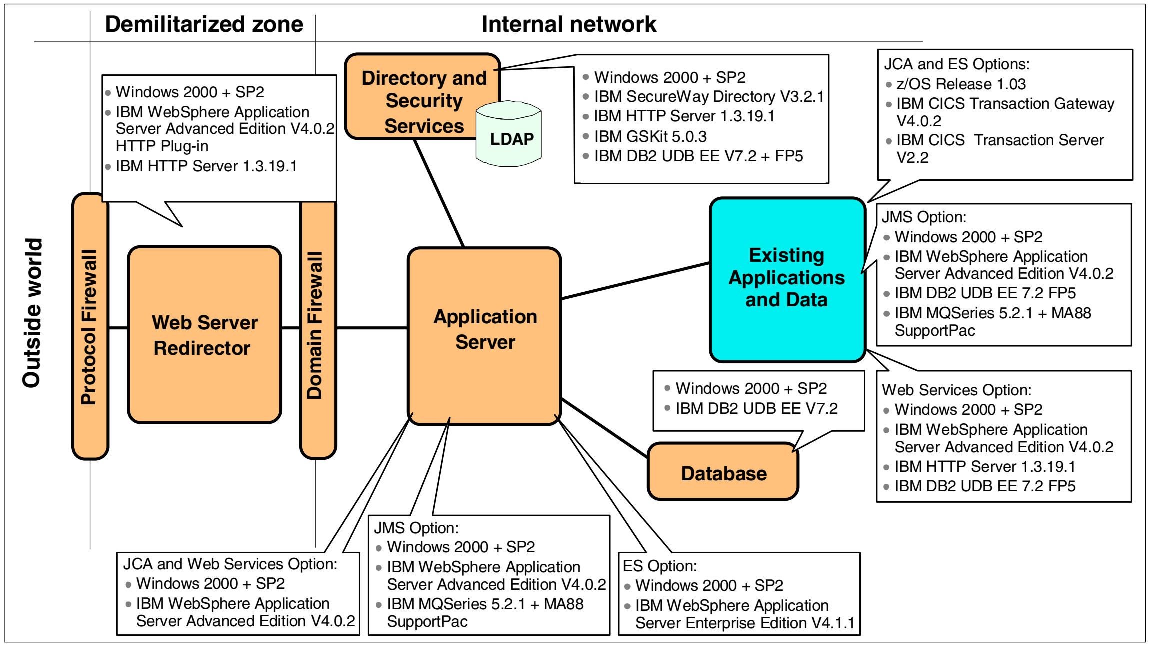

Demilitarized zone
Internal network
Outside world
Protocol Firewall
Web Server Redirector
Domain Firewall
Application Server
Directory and Security Services
LDAP
Existing Applications and Data
Database
• Windows 2000 + SP2
• IBM WebSphere Application Server Advanced Edition V4.0.2 HTTP Plug-in
• IBM HTTP Server 1.3.19.1
• Windows 2000 + SP2
• IBM SecureWay Directory V3.2.1
• IBM HTTP Server 1.3.19.1
• IBM GSKit 5.0.3
• IBM DB2 UDB EE V7.2 + FP5
JCA and ES Options:
• z/OS Release 1.03
• IBM CICS Transaction Gateway V4.0.2
• IBM CICS Transaction Server V2.2
JMS Option:
• Windows 2000 + SP2
• IBM WebSphere Application Server Advanced Edition V4.0.2
• IBM DB2 UDB EE 7.2 FP5
• IBM MQSeries 5.2.1 + MA88 SupportPac
• Windows 2000 + SP2
• IBM DB2 UDB EE V7.2
Web Services Option:
• Windows 2000 + SP2
• IBM WebSphere Application Server Advanced Edition V4.0.2
• IBM HTTP Server 1.3.19.1
• IBM DB2 UDB EE 7.2 FP5
JCA and Web Services Option:
• Windows 2000 + SP2
• IBM WebSphere Application Server Advanced Edition V4.0.2
JMS Option:
• Windows 2000 + SP2
• IBM WebSphere Application Server Advanced Edition V4.0.2
• IBM MQSeries 5.2.1 + MA88 SupportPac
ES Option:
• Windows 2000 + SP2
• IBM WebSphere Application Server Enterprise Edition V4.1.1

Figure 1-11 Directly Integrated Single Channel application pattern: Product mapping

1.2.5 Review guidelines and related links

The Application patterns, Runtime patterns, and Product mappings are intended to guide you in defining the application requirements and the network layout. The actual application development has not been addressed yet. The patterns Web site provides guidelines for each Application pattern that include techniques for developing, implementing, and managing the application based on the following:

- Design guidelines instruct you on tips and techniques for designing the applications.
- Development guidelines take you through the process of building the application, from the requirements phase all the way through the testing and rollout phases.
- System management guidelines address the day-to-day operational concerns, including security, backup and recovery, application management, etc.
- Performance guidelines give information on how to improve the application and system performance.

1.3 Summary

The IBM Patterns for e-business are a collective set of proven architectures. This repository of assets can be used by companies to facilitate the development of Web-based applications. They help an organization understand and analyze complex business problems and break them down into smaller, more manageable functions that can then be implemented.

2

Edge of network Runtime patterns for high availability

This chapter will introduce you to the principles of high availability and some basic high availability Runtime patterns.

We will point out the specialties of each pattern and show how your site can grow to fulfill its needs. Once you are more familiar with these patterns, you will use them as building blocks to design more complex enterprise environments. Applying these patterns, the site can evolve rather than being rebuilt for each change. As in e-business, there is only one constant: change.

2.1 High availability principles

The basic principles of high availability (HA) are very simple. You must consider the possible failure of each component in the architecture of your solution. This will have an impact on your decisions all the way down from the architecture to the design, technology, topology, and products used to build the overall solution.

Once the solution as a whole is defined to meet high availability requirements, you must identify dependencies on which you may, or may not have short-term influence. Your Internet Service Provider (ISP) is an example of this. If your connection to the Internet goes down, all you can do is to wait for it to come back. Dependence on a single ISP is a potential high availability bottleneck. Planning and designing a configuration using multiple ISPs for high availability requires a considerable amount of time to arrange. You must consider and plan for this well in advance.

The skills of your staff and availability of vendor support are crucial assets in supporting the high availability requirements of your configuration.

2.1.1 High availability rules for Runtime environments

In this chapter, we will give you simple rules and guidelines when planning for high availability of your runtime environment and for verifying existing high availability configurations.

There are three basic rules:

1. Every entity must be redundant.
2. Each entity must be monitored for failure.
3. A failed entity must not receive any work.

There is a set of high availability terms that relate to the rules listed above in defining “the anatomy of a failure”. These terms are:

- Detect - continually monitor your configuration for component failures or problems
- Recover - use of redundant entities to continue processing during component failure
- Decide - make a decision to not send work to a failed entity and then decide to send work to that entity after repair
- Repair - fixing the problem component and bringing it back online

2.1.2 Redundancy

In order to make entities useful for high availability, they must be grouped together for redundancy. There are three major groups (see Figure 2-1):

- **High availability pair**

  Some products have the capability to act in pairs, complying with all three high availability rules. There are two kinds of pairs. Primary/Backup, and Peer pairs.

  With the Primary/Backup type, one does the work (Primary role) while the other one monitors it (Backup or standby role). The Backup takes over in case of a detected failure. For example High Availability Cluster Multiprocessing - HACMP (AIX) or Solaris Cluster (SUN) achieve this at an operating system level.

  Another example is WebSphere Edge Server Load Balancer in high availability (HA) mode. Here, it is done at the application level.

  In a peer high availability configuration, both members are actively working and still monitoring each other, taking over the work from its peer in case of a detected failure. WebSphere Edge Server Load Balancer in mutual HA mode works in this way.

- **Cluster**

  Most products that are grouped into a cluster are not capable of monitoring their peers. Usually, they are not even aware of being grouped. They need a third party to distribute requests between them and to comply with high availability rules 2 and 3 listed above. Each member of a cluster must be able to serve identical requests. Members of a cluster are typically capable of serving multiple requests in parallel (multithreaded). So they can have different levels of load assigned to them. Adding members to a cluster and deleting them from the cluster is an administrative task that cannot be done on the distributor instance’s own initiative.

- **Pool**

  If the members of a group are threads, like servlet instances or JDBC connection instances, it is typically referred to as a *pool*. Like members of a cluster, they require a third party to distribute requests and to comply with high availability rules 2 and 3 listed above. In this example of a group, the distributing instance can create and delete threads in the pool to adjust to the overall load on the pool.

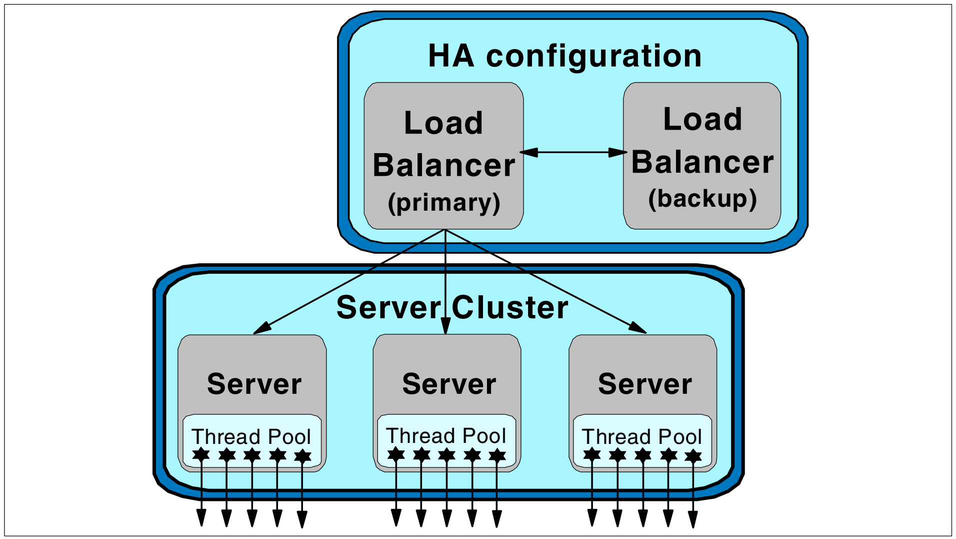


Figure 2-1 HA Pair, Cluster and Pool

2.1.3 Request distribution

To distribute requests to the members of a cluster or pool, there must be an entity that is aware of all members. This entity must address high availability rules 2 and 3. If a distributing entity is enhanced with the capability to measure the load on the cluster members and distribute the load equally among them, it is referred to as a *load balancer*.

There are two distinct configurations using load balancers, the cluster configuration and the high availability configuration.

Cluster configuration

In a cluster configuration, you have multiple entities that perform redirection or basic load balancing functions that are not aware of each other. Therefore, they are grouped in clusters themselves. To comply with high availability rules 2 and 3, these entities would need a third party. Refer to Figure 2-2.

Each entity has the following role:

- Monitor members of a downstream cluster and detect a failure
- Balance requests between the downstream servers in a given cluster

The advantages of a configuration like this are security and separation of processing functions. The security aspect allows you to place the application server behind a domain firewall (not illustrated in Figure 2-2, but shown in later figures). The separation of processing functions allows the machine selection to be made to suit the appropriate processing request (for example Web serving as opposed to application serving).

Examples of technologies that would fit into this category are:

- Caching proxy node enhanced with a content-based routing component (for example, WebSphere Edge Server CBR component)
- Web server redirector node (for example, WebSphere Application Server Plug-in)
- Java Virtual Machine (servlet pool, JDBC connection pool, etc.)
- Stand-alone Load Balancer node (for example, WebSphere Edge Server Network Dispatcher)

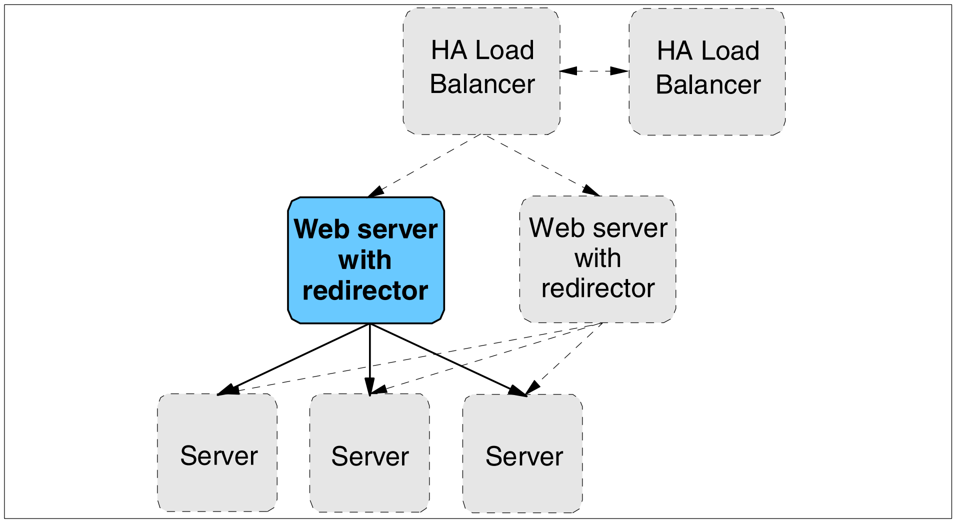


Figure 2-2 Cluster configuration

High availability configuration

In a high availability configuration, the load balancers act in pairs, capable of complying with all three high availability rules. Each load balancer has the following role:

- Capability to monitor its peer load balancer and detect a failure
- Capability to take over the balancing task from its peer load balancer
- Capability to monitor members of a downstream load balancer cluster and detect a failure
- Capability to balance requests between the downstream servers in a given load balancer cluster

An example of this would be the WebSphere Edge Server Load Balancer in high availability mode.

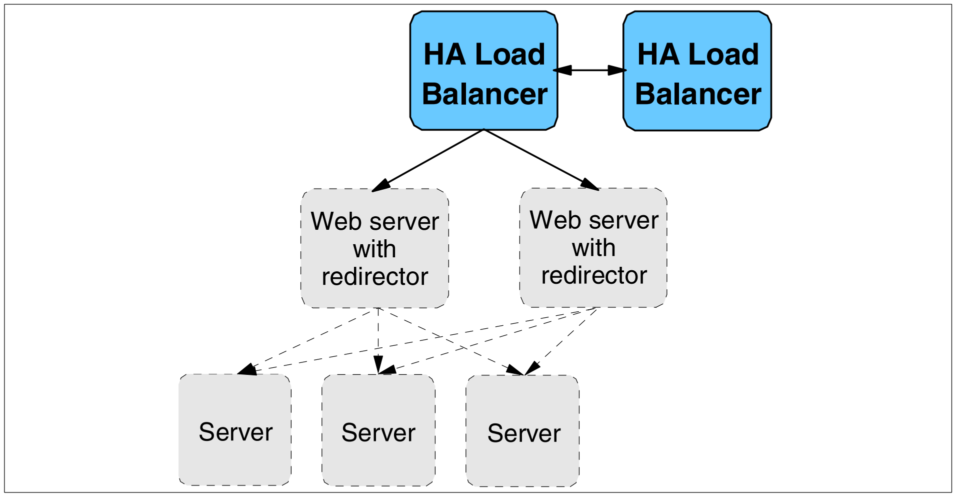


Figure 2-3 Load balancer high availability configuration

2.2 Runtime node types and tiers

The following sections define the types of nodes used to describe runtime patterns and the different locations (tiers) they can occur within a network.

2.2.1 Runtime nodes

To understand the Runtime patterns for high availability, you will need to review the following node definitions.

Public Key Infrastructure node (PKI)

PKI is a collection of standards-based technologies and commercial services to support the secure interaction of two unrelated entities (for example, a public user and a corporation) over the Internet. In the context of the topologies defined in this redbook, PKI supports the authentication of the server to the browser client, using the SSL protocol.

Domain Name System (DNS) node

The DNS node assists in determining the physical network address associated with the symbolic address (URL) of the requested information. The DNS is that of the Internet service provider, although DNS is implemented on the accessed site also.

User node

The user node is most frequently a personal computing device (PC, etc.) supporting a commercial browser, for example, Netscape Navigator or Internet Explorer. The level of the browser is expected to support SSL and some level of DHTML. Increasingly, designers should also consider that this node may be a pervasive computing device, such as a Personal Digital Assistant (see Mobile clients below).

Load balancer node

The load balancer node provides horizontal scalability by dispatching HTTP requests among several, identically configured Web server nodes. Using a load balancer node may introduce the need to ensure session affinity.

A variation of the load balancer node is the high availability load balancer node. This node type introduces a backup load balancer machine that remains ready at all times to take over load balancing tasks should the primary load balancer machine fail. A variation of this is mutual high availability which allows the two load balancer machines to both be active and standby for each other.

Web Application Server node

A Web Application Server node is an application server that includes an HTTP server (also known as a Web server) and is typically designed for access by HTTP clients and to host both presentation and business logic.

The Web Application Server node is a functional extension of the informational (publishing-based) Web server node. It provides the technology platform and contains the components to support access to both public and user-specific information by users employing Web browser technology. For the latter, the node provides robust services to allow users to communicate with shared applications and databases. In this way, it acts as an interface to business functions, such as banking, lending, and HR systems.

This node would be provided by the company, on company premises, or hosted inside the enterprise network and inside a Demilitarized Zone (DMZ) for security reasons. In most cases, access to this server would be in secure mode, using services such as SSL or IPSec.

In the simplest design, this node can provide the management of hypermedia documents and diverse application functions. For more complex applications or those demanding stronger security, it is recommended that the application be deployed on a separate Web Application Server node inside the internal network.

Data that may be contained on the node includes:

- HTML text pages, images, multimedia content to be downloaded to the client browser
- JavaServer Pages
- Application program libraries, for example, Java applets for dynamic downloading to client workstations

Directory and security services node

The directory and security services node supplies information on the location, capabilities and various attributes (including user ID/password pairs and certificates) of resources and users, known to this Web application system. The node may supply information for various security services (authentication and authorization) and may also perform the actual security processing, for example, to verify certificates. The authentication in most current designs validates the access to the Web Application Server node part of the Web server, but it can also authenticate for access to the database server node.

Protocol firewall and domain firewall nodes

Firewalls provide services that can be used to control access from a less trusted network to a more trusted network. Traditional implementations of firewall services include:

- Screening routers (the protocol firewall in this design)
- Application gateways (the domain firewall)

The two firewall nodes provide increasing levels of protection at the expense of increasing computing resource requirements. The protocol firewall is typically implemented as an IP router, while the domain firewall is a dedicated server node.

Database server node

The database server node provides a persistent data storage and retrieval service in support of transactional interactions. The data stored is relevant to the specific business interaction, for example, bank balance, insurance information, current purchase by the user, etc.

It is important to note that the mode of database access is perhaps the most important factor determining the performance of this Web application, in all but the simplest cases. The recommended approach is to collapse the database accesses into a single call or very few calls. This can be achieved via coding and invoking stored procedure calls on the database.

Shared file system node

The timely synchronization of several Web server nodes is achieved by using a shared file system as the content storage and capitalizing on the replication capability of this technology.

2.2.2 Network tiers

Within a typical Runtime pattern, the network is divided into three parts or tiers:

- Outside world
- Demilitarized Zone (DMZ)
- Internal network

Outside world

The first tier is the outside world (Internet). The outside world contains nodes such as the public key infrastructure, domain name servers, and of course user nodes. User nodes are responsible for accepting and validating the user input, communicating the user inputs to the e-business Web server node, and presenting the results received from the Web server node to the user. User nodes typically use HTTP to communicate with the Web Application Server.

Demilitarized Zone (DMZ)

The second tier is known as a Demilitarized Zone. The first and second tiers are separated by a protocol firewall. The DMZ is a network area that is exposed to an untrusted outside world (Internet), and is considered a high-risk zone for attacks. No confidential material should reside on any server in a DMZ in an unprotected form. Only approved ports should be opened between the first and second tier.

Internal network

The internal network (intranet), or third tier, is protected by an additional firewall called a domain firewall. No direct traffic from the first tier is allowed to the third tier. The intranet is considered a safe zone for applications that obtain confidential material.

In this book, we concentrate on high availability and high performance guidelines and scenarios as they apply to nodes in the DMZ for Runtime patterns.

2.3 Runtime patterns for high availability

Runtime patterns are shown in graphical form using standard node types to identify the function represented by that node. Most patterns will consist of a core set of common nodes, with the addition of one or more nodes unique to that pattern.

2.3.1 Application patterns

Runtime patterns and the associated node types are chosen based on the Application pattern selected to address a specific business need. Recall from 1.2.2, “Selecting Application patterns” on page 12 that in some cases, many different Application patterns may be available to satisfy the Business pattern. For our Runtime pattern examples, we will use the Stand-alone Single Channel Application pattern.

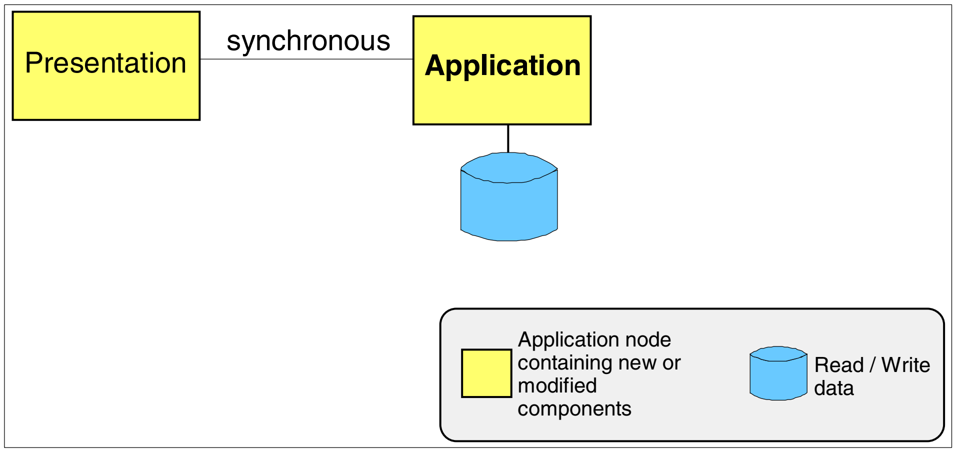


Figure 2-4 Stand-alone Single Channel application pattern

This application pattern provides for stand-alone applications that have no need for integration with existing applications or data. It assumes one delivery channel, most likely a Web client, although it could be something else. It consists of a presentation tier that handles all aspects of the user interface, and an application tier that contains the business logic to access data from a local database. The communication between the two tiers is synchronous. The presentation tier passes a request from the user to the business logic in the Web application tier. The requests is handled and a response sent back to the presentation tier for delivery to the user.

2.3.2 Basic Runtime pattern

Figure 2-5 shows the base Runtime pattern for the Self-Service Stand-alone Single Channel application pattern. We will use it as the standard application pattern for the high availability patterns and variations in order to show the additional nodes you will need for a each high availability configuration. This does not mean that you would not reproduce this configuration in other application patterns. These same principles can be used in many Business and Runtime patterns.

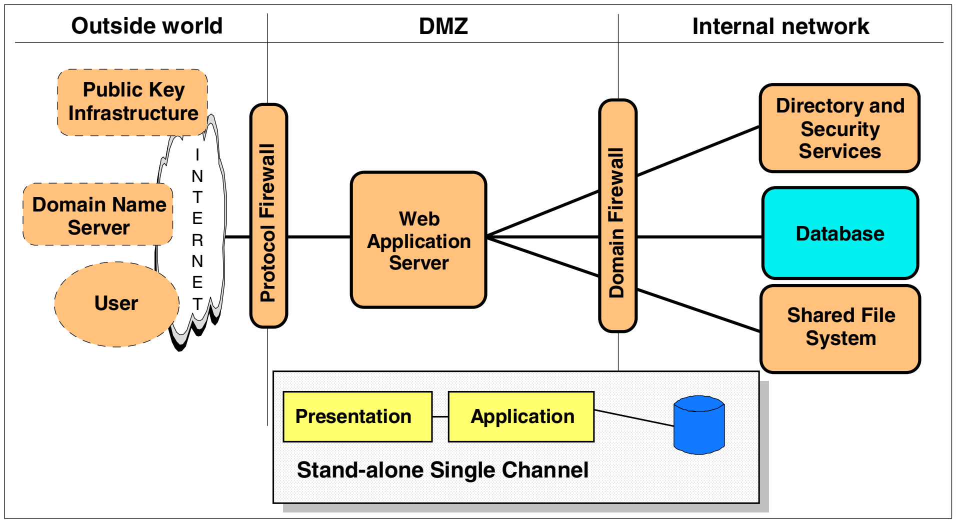


Figure 2-5 Basic Runtime pattern

Both the presentation and application tier are located in the Web Application Server. The Web Application Server node itself is located between two firewalls. This semi-secure area is referred to as the DMZ (Demilitarized Zone). The Web Application Server node also provides authentication and authorization services through the use of a common directory services node in the secure, or internal network behind a second firewall. Application data and application server configuration data is also kept on a database node behind the second firewall.

2.3.3 Basic Runtime pattern variation 1: Single load balancer

In this variation, multiple Web Application Servers are running an application. A single load balancer distributes incoming requests for an application among the Web servers on the Web Application Server nodes.

A load balancer balances incoming traffic among a cluster of back-end servers (that serve the same content) through a combination of load balancing and management software. It can detect a failed back-end server and forward traffic to other servers within the same cluster of back-end servers. When a failed back-end server is brought back online, the load balancer will automatically resume forwarding traffic to it. All client requests received by the load balancer machine are directed to the “best” back-end server in a cluster according to weights that can be set dynamically or fixed. Dynamic weights can be adjusted using advisors to gather information on the health and load of the back-end servers. For a more detailed explanation of the load balancer function as implemented by IBM WebSphere Edge Server, refer to 4.2, “Load Balancer” on page 62.

For this variation, you need the nodes in the basic Runtime pattern and at least two additional nodes:

- One load balancer node
- One or more additional Web Application Server nodes

This Runtime pattern represents an initial step in your effort to provide high availability for the site. Although scalability and high availability for the application are achieved through the use of a load balancer node in front of a cluster of content server nodes, it does not provide failover for the load balancer node itself.

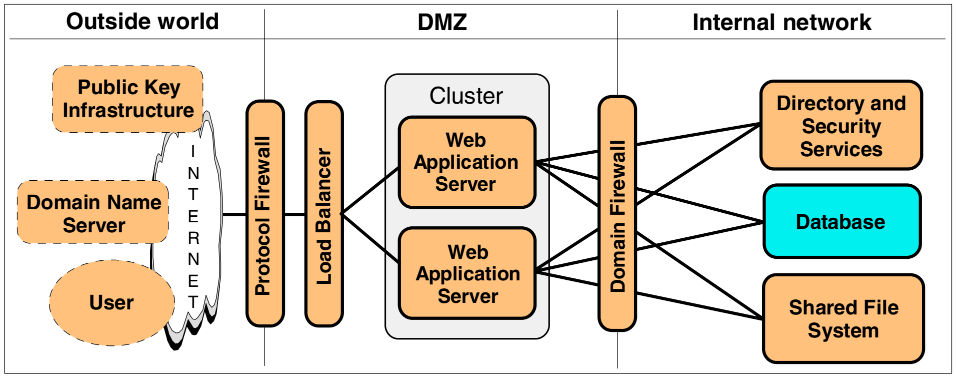


Figure 2-6 Runtime variation 1: Single load balancer

From a security point of view, the load balancer node is neutral. It does not affect the security of your site as long as you do not forward unauthorized requests directly into your internal network.

Note: When designing your runtime topology, be careful not to forward unauthenticated requests from the Internet directly through the domain firewall into your private network. If you have files which do not need to be protected, place the server in the DMZ.

We recommend that you perform an address and/or port change through the NAT/NAPT forwarding method before forwarding requests into your private network. Some companies security policies even demand a protocol change between the first and the second firewall.

Benefits

This runtime topology offers the following benefits:

- Very simple installation and configuration
- Very little delay for the forwarding task if the MAC forwarding method is used
- High availability and failover for the Web Application Server nodes is provided
- Horizontal scalability is possible by simply adding more servers to the content server load balancer cluster

Limitations

This runtime topology has the following limitations:

- The load balancer node still represents a single point of failure

Product mapping

Product mapping correlates real products with one or more runtime nodes. In our high availability scenarios, we will document only the products implemented on nodes in the DMZ since this is where we will demonstrate our high availability alternatives.

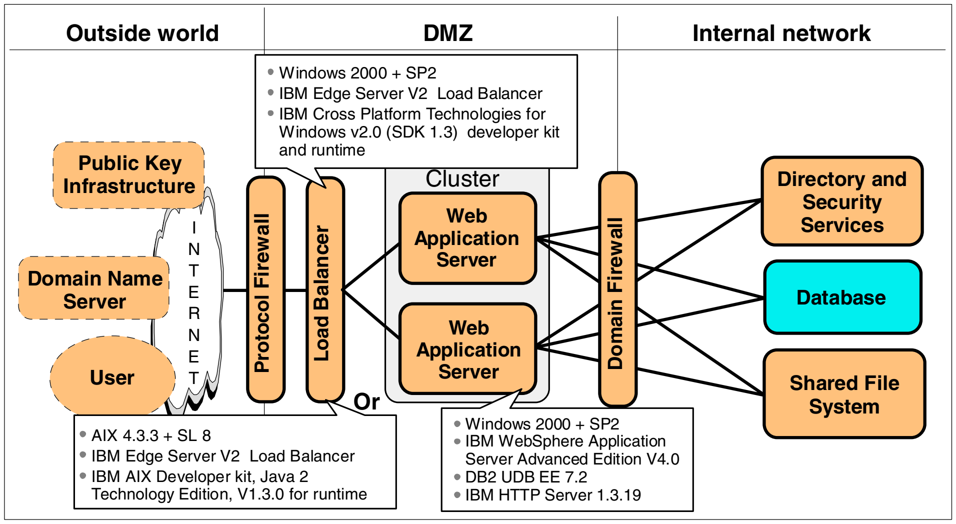

Outside world
DMZ
Internal network
Windows 2000 + SP2
IBM Edge Server V2 Load Balancer
IBM Cross Platform Technologies for Windows v2.0 (SDK 1.3) developer kit and runtime
Public Key Infrastructure
Domain Name Server
User
INTERNET
Protocol Firewall
Load Balancer
Cluster
Web Application Server
Web Application Server
Domain Firewall
Directory and Security Services
Database
Shared File System
Or
AIX 4.3.3 + SL 8
IBM Edge Server V2 Load Balancer
IBM AIX Developer kit, Java 2 Technology Edition, V1.3.0 for runtime
Windows 2000 + SP2
IBM WebSphere Application Server Advanced Edition V4.0
DB2 UDB EE 7.2
IBM HTTP Server 1.3.19

Figure 2-7 Single load balancer product mapping

In Figure 2-7, we list two possible product configurations for the load balancer node. One possible configuration uses Windows 2000 as the base operating system while the other uses AIX V4.3.3 as the base. In both cases, IBM WebSphere Edge Server V2 requires Java 1.3 as a prerequisite. Java 1.3 ships with WebSphere Edge Server for Windows and Linux. For AIX, the Java 1.3 runtime environment can be obtained at:

`http://www.ibm.com/developerworks/java/jdk/index.html`

IBM WebSphere Edge Server

The load balancer node is implemented using Load Balancer Dispatcher component of IBM WebSphere Edge Server V2.0. The Load Balancer component is configured so the two WebSphere Application Servers comprise one load balanced cluster. Requests can be distributed round-robin or based on load using advisors. The Media Access Control (MAC) forwarding method of Load Balancer can be used in this configuration (see 4.2.2, “Media Access Control (MAC) forwarding” on page 65 for additional information). However, both Network Address Translation (NAT) and Network Address Port Translation (NAPT) forwarding methods could be used. Enhanced recognition of available CPU and memory resources on content servers using advisors or metric server agents is provided with the Load Balancer. Session affinity to a specific Web server can be maintained using the sticky time affinity feature or the active cookie support of the Load Balancer. However, the use of application server clones using cookie-based server affinity is also an option.

SSL connections from the client to the load balancer will be tunneled to the Web Application Servers.

WebSphere Application Server

The WebSphere Application Servers are configured with the same applications available on both servers. This can be accomplished using clones. Cookie-based server affinity is provided by the WebSphere Application Servers

2.3.4 Basic Runtime pattern variation 2: Load balancer hot standby

This is the standard high availability solution for the load balancer nodes themselves. This pattern removes the single point of failure from the load balancer node by adding a second node with the same function. The first node will be the active, working load balancer. It is the primary load balancer. The second one keeps track of the balancing information and is ready to take over the work of the primary node at any moment. It is the standby load balancer.

For additional information on Load Balancer High Availability support, see 4.2.5, “Load Balancer High Availability feature” on page 71.

For this variation you need at least three nodes in addition to the basic Runtime pattern:

- Two load balancer nodes
- One or more additional Web Application Server nodes

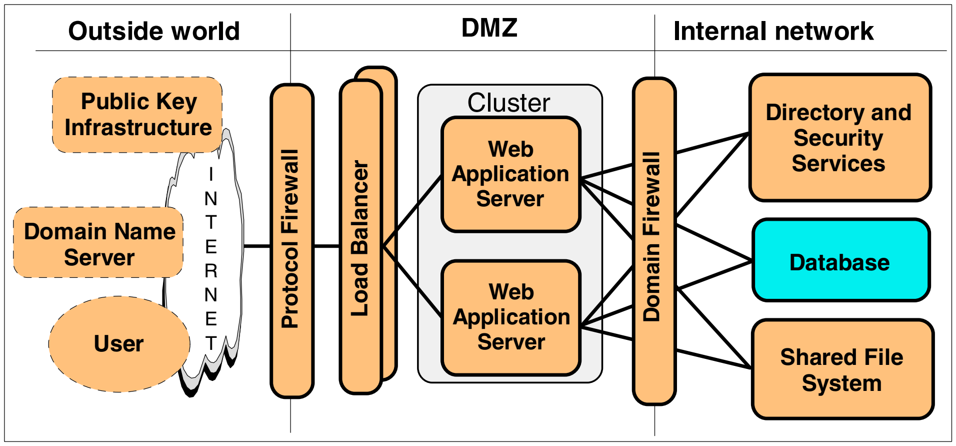


Figure 2-8 Basic Runtime pattern variation 2: Load balancer hot standby

Benefits

This variation offers all of the benefits listed for variation 1, plus:

- High availability and failover is provided for the load balancer nodes

Disadvantages

This variation has the following disadvantage:

- One server is in hot standby mode, not doing any productive work

Product mapping

There are two product mapping options available for the load balancer hot standby configuration.

Product mapping option 1

In this product mapping, two separate load balancer machines are used to perform the primary and standby load balancer functions.

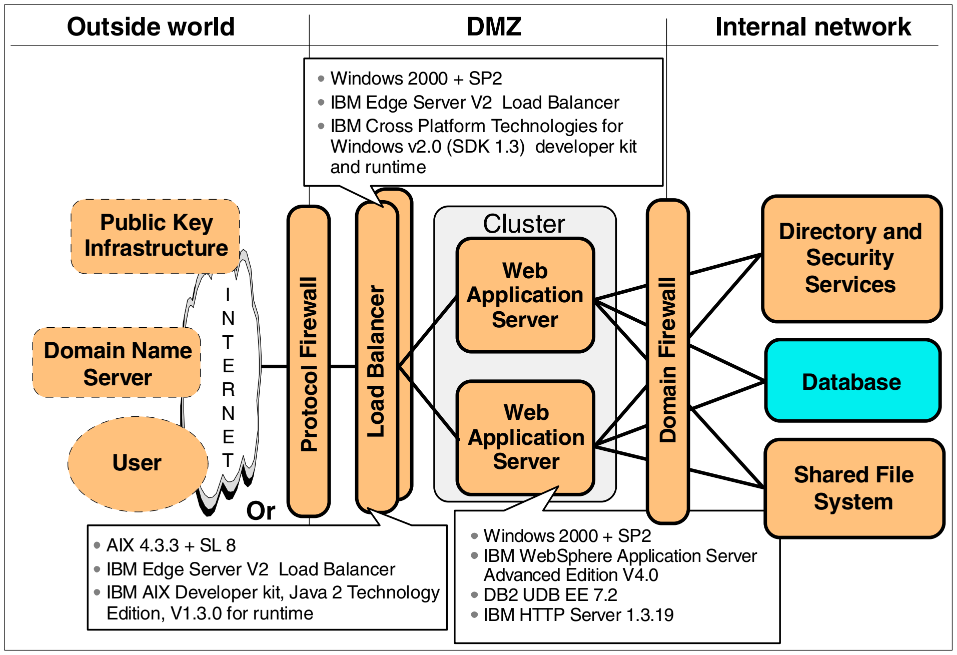

Outside world
DMZ
Internal network
• Windows 2000 + SP2
• IBM Edge Server V2 Load Balancer
• IBM Cross Platform Technologies for Windows v2.0 (SDK 1.3) developer kit and runtime
Public Key Infrastructure
Domain Name Server
User
INTERNET
Or
Protocol Firewall
Load Balancer
Cluster
Web Application Server
Web Application Server
Domain Firewall
Directory and Security Services
Database
Shared File System
• AIX 4.3.3 + SL 8
• IBM Edge Server V2 Load Balancer
• IBM AIX Developer kit, Java 2 Technology Edition, V1.3.0 for runtime
• Windows 2000 + SP2
• IBM WebSphere Application Server Advanced Edition V4.0
• DB2 UDB EE 7.2
• IBM HTTP Server 1.3.19

Figure 2-9 Load balancer hot standby product mapping option 1

The Edge Server Load Balancer components keep in constant contact with each other by exchanging a heartbeat between them. This enables the standby load balancer to take over without losing IP packets.

Product mapping option 2

This product mapping option provides a load balancer hot standby configuration without increasing the need for physical machines.

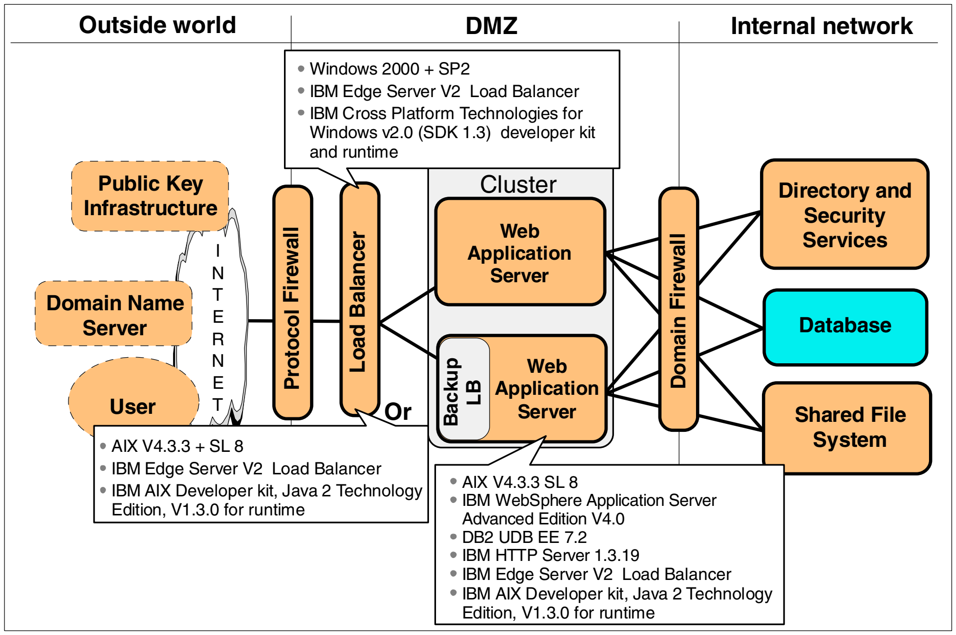

Outside world
DMZ
Internal network
• Windows 2000 + SP2
• IBM Edge Server V2 Load Balancer
• IBM Cross Platform Technologies for Windows v2.0 (SDK 1.3) developer kit and runtime
Public Key Infrastructure
Domain Name Server
User
INTERNET
Protocol Firewall
Load Balancer
Or
Cluster
Web Application Server
Backup LB
Web Application Server
Domain Firewall
Directory and Security Services
Database
Shared File System
• AIX V4.3.3 + SL 8
• IBM Edge Server V2 Load Balancer
• IBM AIX Developer kit, Java 2 Technology Edition, V1.3.0 for runtime
• AIX V4.3.3 SL 8
• IBM WebSphere Application Server Advanced Edition V4.0
• DB2 UDB EE 7.2
• IBM HTTP Server 1.3.19
• IBM Edge Server V2 Load Balancer
• IBM AIX Developer kit, Java 2 Technology Edition, V1.3.0 for runtime

Figure 2-10 Load balancer hot standby product mapping option 2

IBM Edge Server allows you to run a backup Load Balancer on one of the Web Application Server machines. This is referred to as *collocation*. Collocation of the WebSphere Edge Server Load Balancer component is supported on AIX, Linux and Solaris platforms.

In this configuration, the backup load balancer node could be affected by problems on the Web Application Server node. In addition, collocation of the backup load balancer function on a content server machine is not supported on all platforms.

2.3.5 Basic Runtime pattern variation 3: Mutual high availability

This variation demonstrates an advanced way to run two load balancer nodes in a highly available configuration called the *mutual high availability mode*. In the previous patterns, a second load balancer node existed, but was in hot standby mode. It did not perform load balancing unless the primary load balancer node failed.

In a mutual high availability configuration, both load balancer nodes are active and balancing requests for their own configured back-end server clusters. It is possible to configure this feature if you have at least two clusters of back-end servers to be load balanced.

As can be seen in Figure 2-11, each load balancer is configured with its own set of back-end servers to be load balanced. In addition, each load balancer also has connections to, and definitions for, the back-end servers of the other load balancer machine. Each load balancer monitors the health of the other load balancer, and if one detects the failure of the other, it takes over the load balancing function for the failing load balancer machine.

Each load balancer presents a different IP address and DNS name to the Internet. When a failure is detected by one of the load balancer machines, one of the functions it performs during takeover is to dynamically alias the IP address of the failing load balancer machine to its NIC. It will then receive packets destined for the failed machine.

For additional information on mutual high availability, see 4.2.6, “Load Balancer Mutual High Availability feature” on page 73.

For this variation, you need at least three nodes in addition to the Basic Runtime pattern:

- Two load balancer nodes.
- One or more additional Web Application Server nodes.

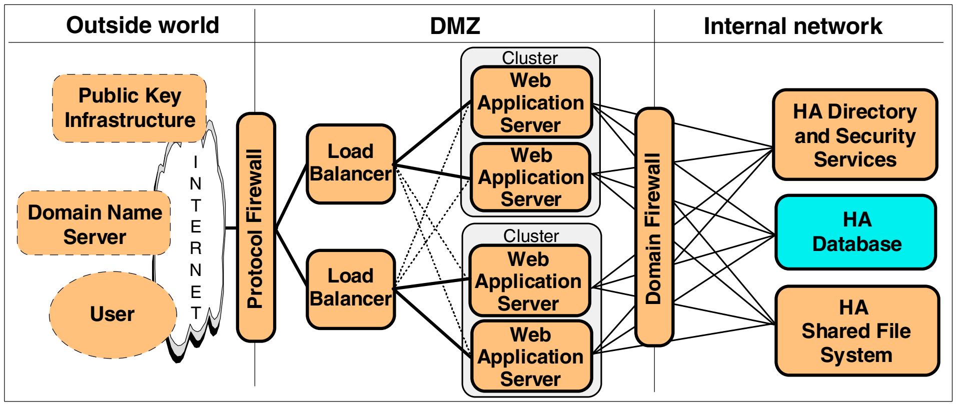


Figure 2-11 Basic Runtime pattern variation 3: Mutual high availability

With mutual high availability, each load balancer node has at least one load balancer cluster assigned to it and hence acts in the role of the primary load balancer for it. Each node also performs the backup load balancer role for the cluster(s) of its partner node.

Benefits:

Additional benefits from previous Runtime patterns are:

- Both load balancer nodes are active and perform load balancing.

Disadvantages:

This variation has the following disadvantage:

- The configuration of mutual high availability is more complex

Product mapping

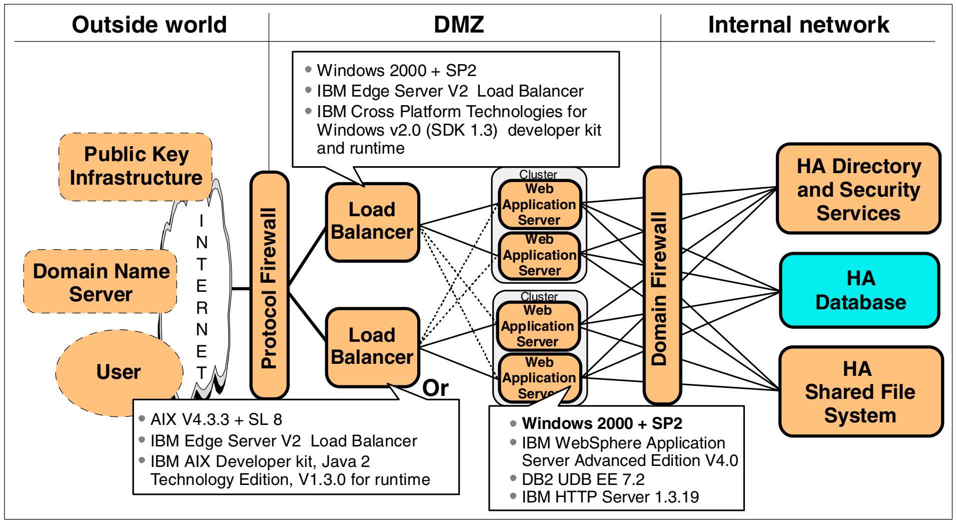


Figure 2-12 Mutual high availability product mapping

Note in Figure 2-12 that one of the load balancer machines is running Windows 2000 while the other machine is running AIX V4.3.3. This is supported, but not recommended because the different processing speeds and capacities between the two machine types could potentially cause performance degradation in a backup situation.

2.3.6 Basic Runtime pattern variation 4: wide area load balancing

This variation introduces load balancing among remote networks. A load balancer node balances requests between remote networks. Load balancer nodes in each local network are members of a wide area cluster and act as both a router between the main load balancer node and the local cluster as well as a load balancer for the local cluster. Such a local cluster could also represent an entire site.

The main load balancer node should always be set up in high availability mode.

For additional information on wide area load balancing, see 4.2.7, “Wide Area Network Dispatcher (WAND)” on page 75.

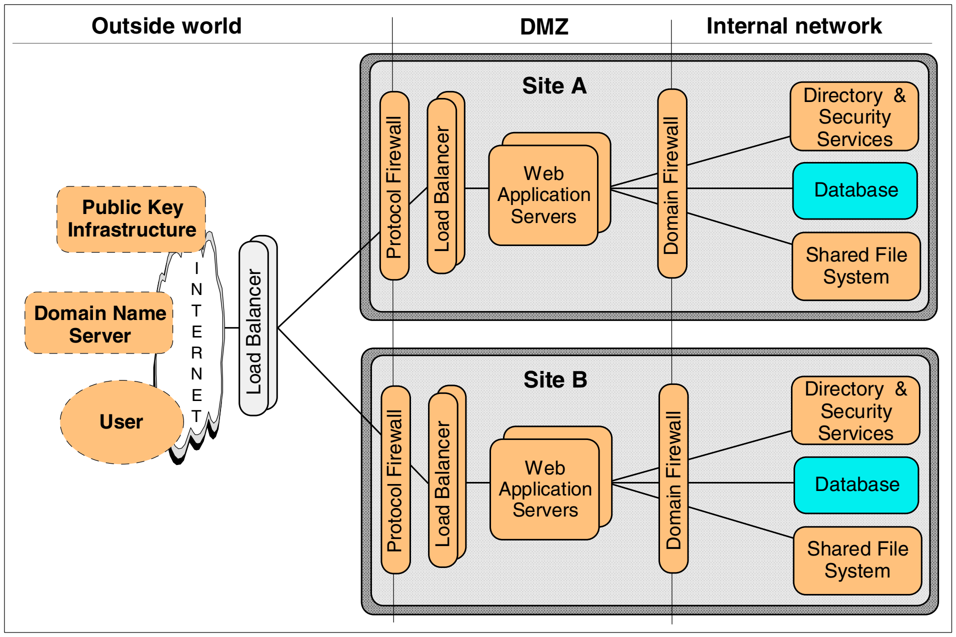


Figure 2-13 Basic Runtime pattern variation 4: Wide area load balancing

Benefits

This variation provides all the advantages of variation 1, plus:

- Provides high availability among remote clusters or even entire sites, but is still simple to implement
- Supports cookie-based server affinity
- Entire sites can be added and removed by simply adding or removing the local load balancer node from the wide area load balancer cluster
- When using MAC forwarding, response data is sent directly to the client

Disadvantages

This variation has the following disadvantage:

- When load balancing entire sites, everything depends on those main load balancer nodes

Product mapping option 1

The product mapping is similar to variation 1. In Figure 2-14, the product mappings for Site A and Site B would correspond to those seen in Figure 2-7 on page 32.

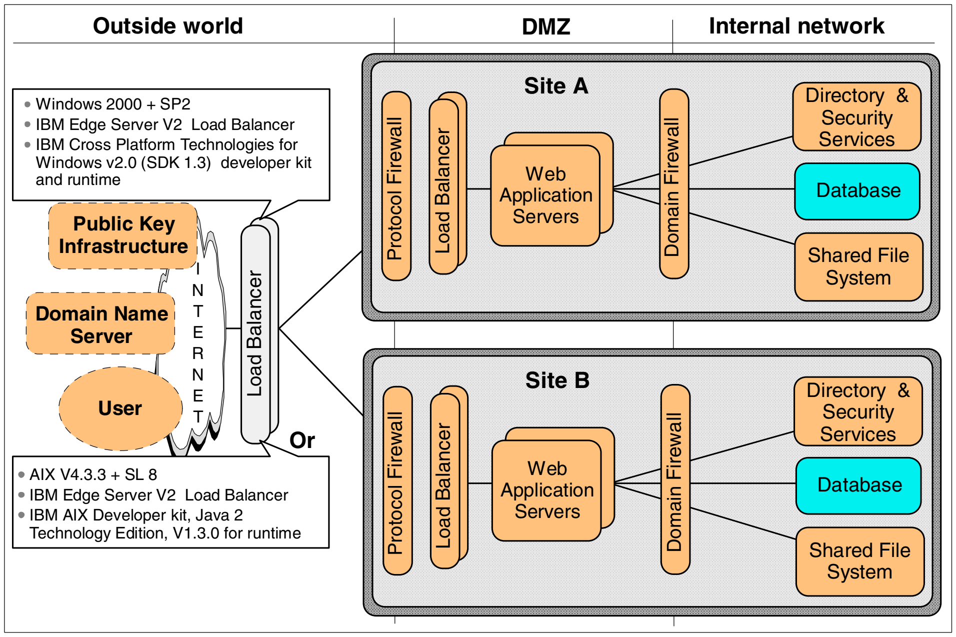


Figure 2-14 Wide area load balancing product mapping option 1

The Dispatcher component of Load Balancer provides a Wide Area Network Dispatcher (WAND) enhancement that offers support for remote servers. A remote server consists of a remote load balancer machine and its locally attached back-end load balanced servers.

A client's request packet can flow from the Internet to a load balancer machine, from there to a geographically remote load balancer machine, then to one of its locally attached back-end servers. The data returned to the client from the back-end server can flow directly back to the Internet and the client.

This allows one load balancer cluster address to support all worldwide client requests while distributing the load to back-end servers around the world. The load balancer machine initially receiving the request can still have local servers attached to it and can distribute the load among its local back-end servers as well as the remote servers.

Product mapping option 2: Site Selector

This variation is an alternative of the wide area load balancing described previously. As in variation 4, remote clusters are load balanced. However, in this instance, the load balancing is done by mapping DNS names to IP addresses, acting as a non-recursive (leaf node) name server.

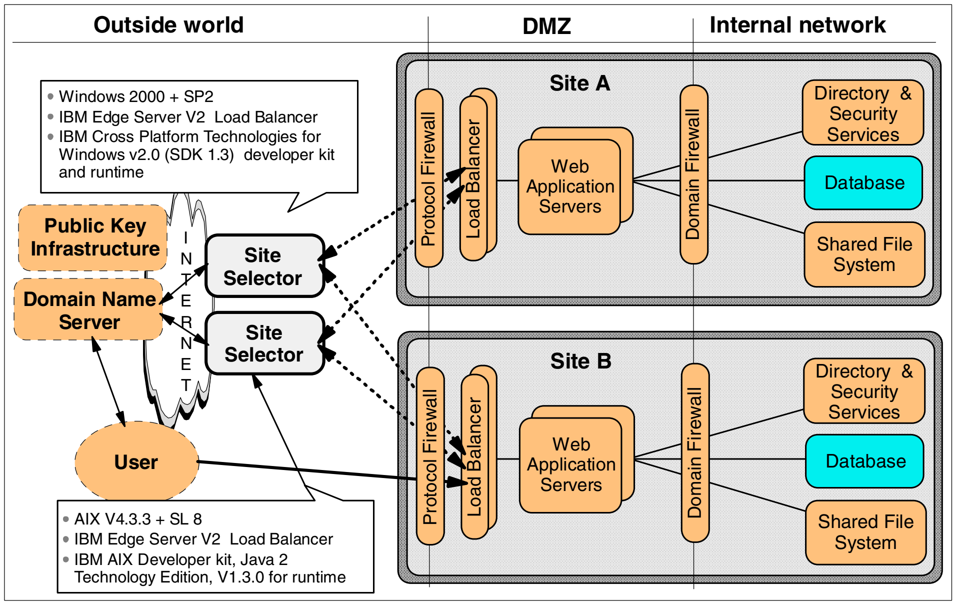

Outside world
DMZ
Internal network
Windows 2000 + SP2
IBM Edge Server V2 Load Balancer
IBM Cross Platform Technologies for Windows v2.0 (SDK 1.3) developer kit and runtime
Public Key Infrastructure
Domain Name Server
INTERNET
Site Selector
Site Selector
User
AIX V4.3.3 + SL 8
IBM Edge Server V2 Load Balancer
IBM AIX Developer kit, Java 2 Technology Edition, V1.3.0 for runtime
Site A
Site B
Protocol Firewall
Load Balancer
Web Application Servers
Domain Firewall
Directory & Security Services
Database
Shared File System

Figure 2-15 Wide area load balancing product mapping option 2

Site Selector is a component of WebSphere Edge Server Load Balancer. It communicates with the Domain Name Server (DNS) to return the appropriate IP address to the client based on feedback from the back-end servers. It selects the “right” cluster or back-end server IP address to satisfy a client name service request. This allows the client to connect directly to the cluster or back-end server for all its communications.

Site Selector can be used in conjunction with advisors and the Metric Server component of Network Dispatcher (refer to 4.2.9, “Metric Server” on page 77) to determine the best server that a client request should be sent to. Additionally, if Site Selector is used with Metric Server, it can make use of a feature called Network Proximity, which allows Site Selector to choose the best server according to the shortest response time of a ping between the servers and the client.

You should note the following considerations:

- DNS query responses might be cached at various places, delaying the change of Site Selector's preferred target servers. Browsers will often cache DNS resolved IP addresses.
- Site Selector depends on DNS servers, for which it acts as a leaf node.

3

Edge of network Runtime patterns for performance

In this chapter, we discuss performance issues and provide some basic high performance Runtime patterns.

The baseline to performance is apparent: you want good response times for your clients accessing your Web site. Today, this is not only a nice thing to have, but an imperative demand in order to survive on the Internet and be successful against your competitors.

3.1 Performance factors

There are two distinct aspects to performance. One is the minimum amount of time needed to process a given task. This is based on the nature of the task and the application design. The other is maintaining the amount of time needed for each task if the number of requests for that task increases.

Performance is the result of three factors that interact with each other:

- Time
- Resources
- Workload

If we look at it in a abstract way, it is a triangular dependency.

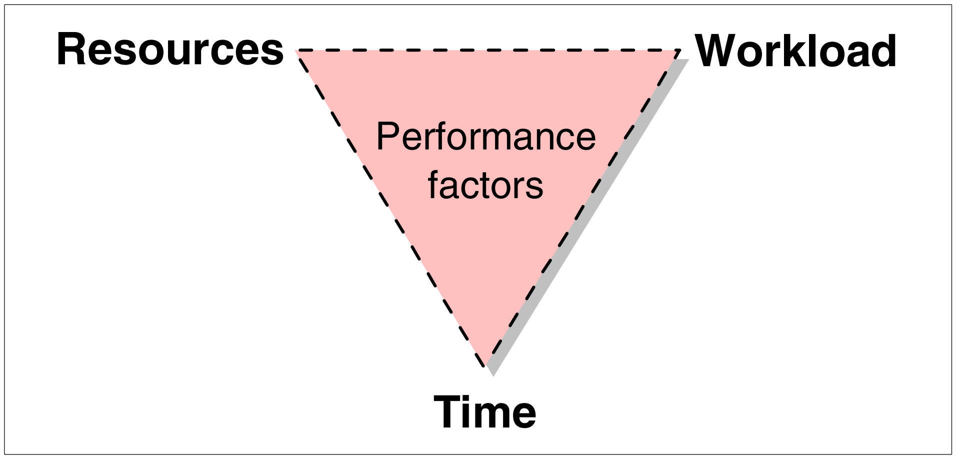


Figure 3-1 Performance factors

Each of the factors is a result of the other two. For a Web site, the time factor is the targeted response time.

Time = Workload \ Resources

To deal with the time factor, we can do three things:

- Increase the resources

  This is the obvious approach to increased throughput. The condition to meet is scalability of both the application and the runtime environment. Scalability of the runtime environment includes high availability. It is achieved through an architecture that allows load balancing and clustering of resources. Scalability of the application is a matter of its design and will not be further discussed here.

- Lower the workload

  Because the time needed for a given task is defined by its nature and the design of the application, we cannot influence it here. But we can try to reuse the product of such a task. Caching is the technique to achieve this goal. Today not only static content can be cached, but also dynamic results of processing tasks.

- Increase the time available

  Although your first thought might be that that is exactly what cannot be done, we can do it in a abstract way. We can split the workload into sub tasks where different functionality is needed and distribute them to specialized instances. In this way, they can be processed in parallel and so the overall time available is increased. This will be limited or enhanced by many factors such as application design, application packaging, runtime topology, skills available, and so forth. Good planning and communication between the business units involved is crucial here.

3.2 Enhancing performance with proxy servers

A proxy server has the ability to act as a relay station for clients. Forwarding requests and receiving the responses makes it an ideal place to cache request results for reuse. This kind of proxy server is known as a caching proxy. This has several benefits:

- The response time to the client requests is reduced
- The load on the content server nodes is reduced
- Less network bandwidth between the content server nodes is used
- By freeing up resources, better scalability is achieved

A proxy server can also be used to establish an additional layer between the client and the content server nodes by hiding their IP addresses and locations, or to intercept requests for authentication. For more information, see Chapter 5, “Security guidelines” on page 89.

Static caching

When a client requests a file from a Web server, the Web server must find the file, read it from disk and send it out to the network. Disk IO operations are among the most time consuming operations for a Web server. Saving such a file in memory and delivering it from there dramatically reduces the response time for all subsequent requests. This technique is known as caching. Today, most Web servers support the use of their own caches. Having redundant Web servers in a cluster results in redundant caching of files. Implementing a caching layer in front of the Web server clusters further improves the cache utilization and cache hit rate.

Dynamic caching
For e-business applications, much of the memory and CPU processing power is used to generate the dynamic presentation of data. Caching the results of those operations is known as dynamic caching. It reduces response time to the client and reduces the load on both the redirector server nodes and the application server nodes. It also reduces the used bandwidth between the content server nodes.

Memory cache
Memory is by far the fastest media to write to and read from. Therefore, this is the preferred cache storage. There are limitations to the amount of memory a server can handle and the cost per megabyte of storage space is much higher than on a disk. Memory cache is preferred for frequently changing data of small to medium size amounts.

Disk cache
For large amounts of data, disk caches are still a reasonable compromise between performance benefit and storage cost. For very large caches (hundreds of gigabytes), disk caching is the proper choice.

Cache arrays
Multiple caches can be grouped together to form an cache of a much larger total size than one server could ever handle. Such a group is called a cache array. Specialized protocols such as Cache Array Routing Protocol (CARP) and Internet Caching Protocol (ICP) have been developed to enable the members of a cache array to communicate with each other, avoiding redundant caching within a cache array.

Proxy chaining
Implementing several layers of proxy server nodes enables the combination of different functionalities to build sophisticated caching architectures.

3.3 Runtime nodes

In addition to the nodes listed in 2.2.1, “Runtime nodes” on page 24, the following nodes will be used:

Caching proxy node

A proxy server node intercepts data requests from a client, retrieves the requested information from content-hosting machines, and delivers that content back to the client. Most commonly, the requests are for documents stored on Web server machines and delivered via the Hypertext Transfer Protocol (HTTP). However, some proxy servers also handle other protocols such as File Transfer Protocol (FTP) and Gopher. The IP address of the content-hosting machines is not seen by the clients.

A caching proxy node stores cacheable content in a local cache before delivering it to the client. Subsequent requests for the same content are served from the local cache which is much faster and does not consume as much content server processing and network bandwidth.

Web server redirector node

In order to separate the Web server node from the application server node, a Web server redirector node (or just redirector for short) is introduced. The Web server redirector is used in conjunction with a Web server. The Web server redirector serves HTTP pages and forwards servlet and 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 DMZ. Static pages can be served from the DMZ by this node.

Application server node

The application server node provides the infrastructure for application logic and may be part of a Web Application Server node. 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 node will run both presentation and business logic. In other situations, it may be used for business logic only.

Presentation server node

A presentation server node provides support to assemble HTML output in response to client requests. It supports running of Java applets and JavaServer Pages in the process of creating completed HTML output pages. It generally does not serve HTTP requests. When used with a Web server redirector, the presentation server node will run the presentation logic while a separate application server node runs the business logic. This allows the presentation server node to reside in the DMZ while maintaining critical business logic and data access in the secure intranet.

3.4 Runtime patterns

We will use the Basic Runtime pattern variation 2 (2.3.4, "Basic Runtime pattern variation 2: Load balancer hot standby" on page 33) as the starting point for the following patterns.

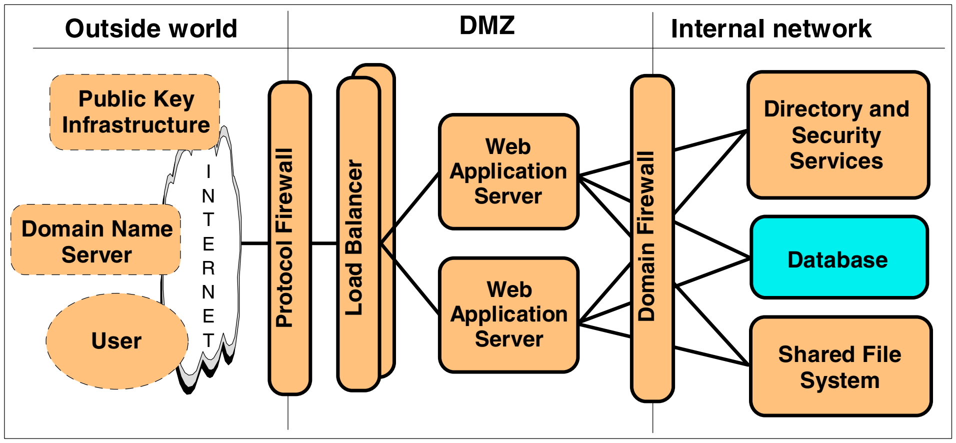


Figure 3-2 Basic Runtime pattern variation 2

3.4.1 Basic Runtime pattern variation 5: Redirectors

For this variation, you need at least two additional nodes:

- Two or more Web server redirector nodes

This pattern splits the Web Application Server node into two functional nodes by separating HTTP server function from the application server function.

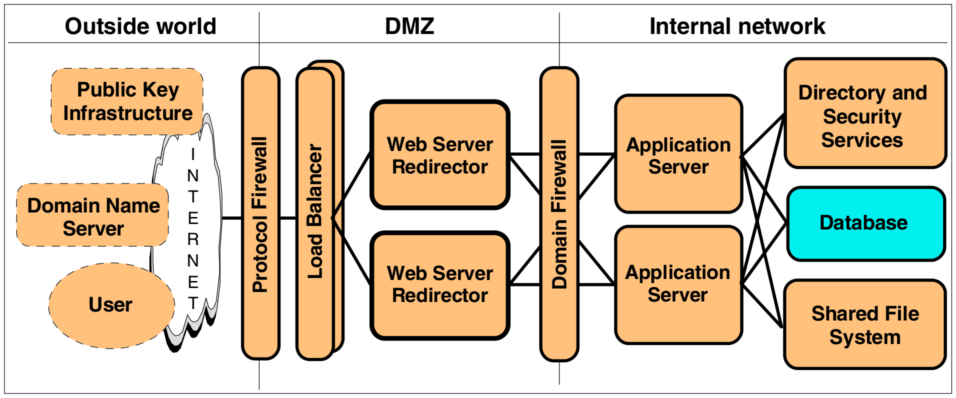


Figure 3-3 Basic Runtime pattern variation 5: Redirectors

The Web server redirector node decides whether the request is being served by the local HTTP server or forwarded to the application server nodes. This pattern moves the application server nodes behind the domain firewall, adding further security.

The Web server redirector node is capable of performing workload management for the requests targeted at the application server nodes in the secure network. In addition, it can perform a port change to match the ports on which the application server nodes are listening. It also takes care of cookie-based server affinity.

Benefits

This variation has the following additional benefits:

- Each node type (Redirector and Application Server) can be configured to the specific tasks
- A domain firewall between the two node types adds additional security
- More flexibility in number of nodes of each type
- Flexibility to not apply security on certain static content

Disadvantages

This variation has the following disadvantages:

- Many configuration changes on the Application Server node call for a redeployment of the plugin-cfg.xml file to all Web server redirector nodes.
- The configuration of the domain firewall requires additional skills; especially if NAT is implemented, IIOP traffic must be tunneled.

Product mapping

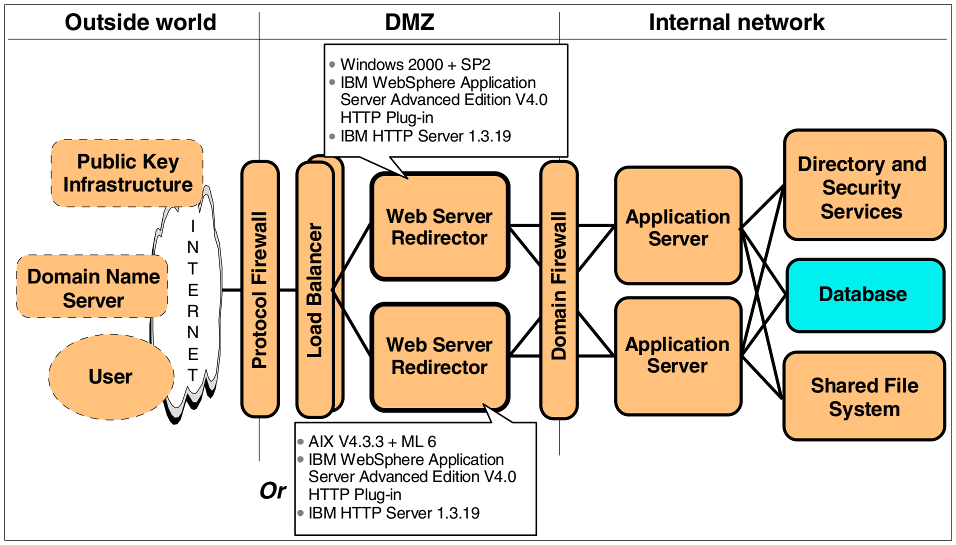


Figure 3-4 Redirector product mapping

The Web server redirector nodes run the IBM HTTP Server with the WebSphere Application Server HTTP plug-in. Again, notice that two redirector nodes are illustrated in Figure 3-4. One redirector node could possibly be sufficient to satisfy all requests. However, high availability considerations recommend using two redirectors with load balancer spraying requests to both of them.

3.4.2 Basic Runtime pattern variation 6: separation

The Runtime pattern we see here further separates the functionality to distinct Application Server nodes. The presentation tasks (Web modules) are run on the Presentation Server behind the domain firewall. The application tasks (EJB modules) run on the Application Server nodes behind the domain firewall in the secure network.

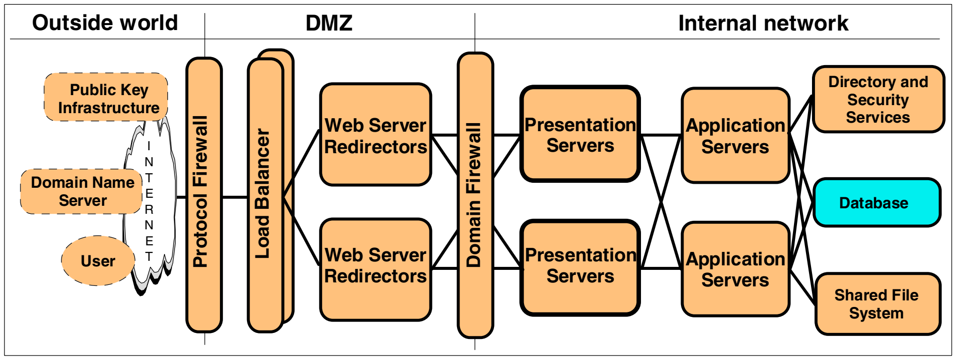


Figure 3-5 Basic Runtime pattern variation 6: separation

With this topology, you would create two (WebSphere) server groups in order to execute the presentation and the application modules on different nodes. One would have clones on the Presentation Server nodes in the secure network running the Web modules. The other one would have clones on the Application Server nodes in the secure network running the EJB modules. In this manner, more repetitive and mundane presentation tasks can be offloaded from the application servers, freeing more processing power for business application workload.

Benefits

This variation has the following additional advantages:

- Flexibility to choose the protocol between the two application server nodes (HTTP, HTTPS, RMI, RMI/IIOP)
- Further separation of presentation and application functionality

Disadvantages

This variation has the following disadvantages:

- Configuration changes on the Presentation Server node may require a manual update of the Web server redirector nodes
- More planning for application deployment is required

Product mapping

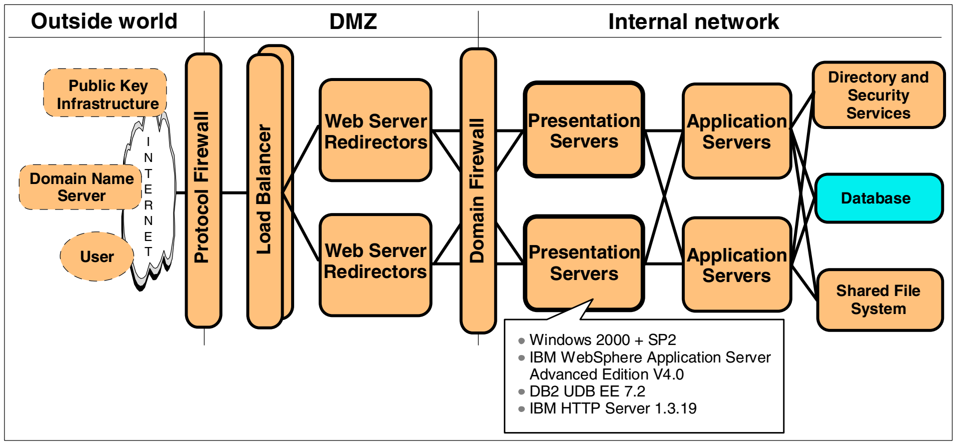


Figure 3-6 Separation product mapping

The WebSphere Application Server V4 HTTP plug-in that runs in the HTTP Redirector nodes can perform a load balancing function to spray client requests between the two presentation server nodes.

3.4.3 Basic Runtime pattern variation 7: caching proxy

Variation 7 introduces a layer of caching proxy server nodes between the load balancer nodes and the Web Application Server nodes. Because a pure caching proxy node cannot balance requests to multiple (identical) proxy target nodes, a second layer of load balancer nodes is introduced as well. The second layer of load balancing can be configured on the two existing load balancer nodes.

For Runtime pattern 7, you need at least two nodes in addition to those in Basic Runtime pattern variation 2:

- Two or more additional caching proxy server nodes

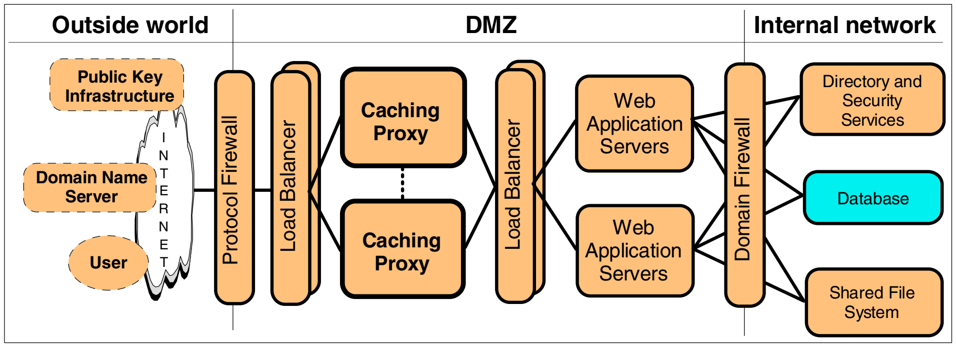


Figure 3-7 Basic Runtime pattern variation 7: Caching proxy

The first load balancer node passes the request for the information to the caching proxy nodes. The caching proxy determines if the requested page is in its cache. If the requested information is in the cache, it is served from there. If it is not in the cache, the request is passed on by the caching proxy to the second load balancer node which forwards the request to the back-end servers.

To minimize the overlap of duplicated content in the local cache of each caching proxy machine, remote cache access (RCA) can be implemented in each caching proxy. RCA allows each proxy to share the contents of its cache with the other proxy servers. This results in bandwidth savings because objects are not fetched multiple times, and the larger combined logical cache yields a higher hit rate. The challenge in efficiently using multiple caching proxy nodes is to have high speed connections between them or, in case of disk cache, a high performance shared file system.

Benefits

Variation 7 offers the following benefits:

- High availability is provided for all node types
- The caching proxy layer reduces response times for static, cachable content
- It reduces the load on the Web server redirector nodes
- Supports SSL encryption between client and Proxy Server node as well as from Proxy Server node to content server node

Limitations

Variation 7 has the following limitation:

- To harvest the benefit of cache arrays, high speed connections or a high performance shared file system between the caching proxy server nodes is necessary

Product mapping

As stated previously, product mapping correlates real products with one or more runtime nodes. In our scenarios, we will document only the products implemented on nodes in the DMZ, since this is where we will demonstrate our alternatives.

In the following product mapping visuals, we will document only software that is new or modified from previous product mapping configurations shown in Figure 2-7 on page 32.

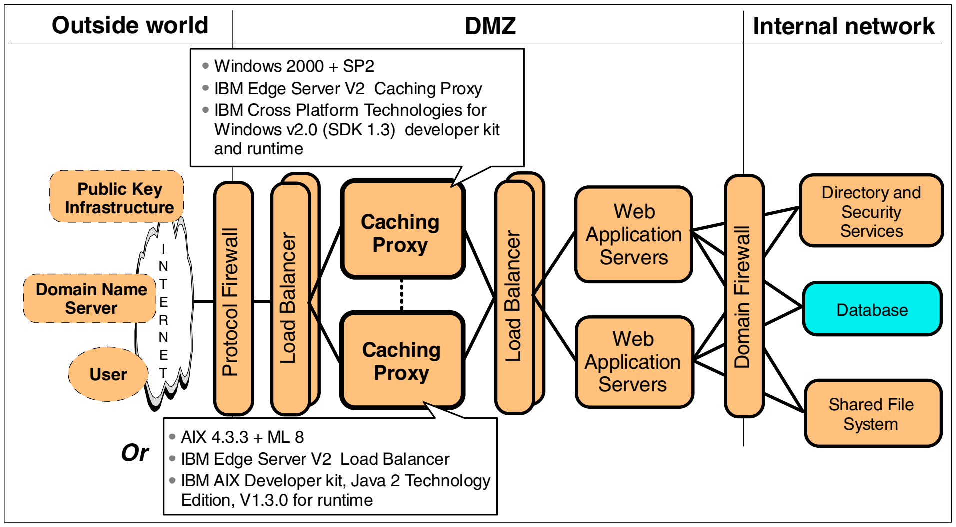

Outside world
DMZ
Internal network
Windows 2000 + SP2
IBM Edge Server V2 Caching Proxy
IBM Cross Platform Technologies for Windows v2.0 (SDK 1.3) developer kit and runtime
Public Key Infrastructure
Domain Name Server
User
INTERNET
Protocol Firewall
Load Balancer
Caching Proxy
Caching Proxy
Load Balancer
Web Application Servers
Web Application Servers
Domain Firewall
Directory and Security Services
Database
Shared File System
Or
AIX 4.3.3 + ML 8
IBM Edge Server V2 Load Balancer
IBM AIX Developer kit, Java 2 Technology Edition, V1.3.0 for runtime

Figure 3-8 Caching proxy product mapping

Note that two caching proxy nodes are illustrated in Figure 3-8. This is a high availability configuration with one caching proxy node able to perform all caching functions if the other node is unavailable. One possible configuration uses Windows 2000 as the base operating system while the other uses AIX V4.3.3 as the base. In both cases, IBM WebSphere Edge Server V2 requires Java 1.3 as a prerequisite.

IBM WebSphere Edge Server

The caching proxy nodes are implemented using the Caching Proxy component of IBM WebSphere Edge Server V2.0. The caching proxy component can be configured so that cached pages are saved in only one caching proxy node. At the same time, if one caching proxy node becomes unavailable, the remaining node can assume the caching function for all cached pages.

Note in Figure 3-8 that there are two sets of high availability load balancer nodes. Both of these load balancer instances can be configured on the same pair of machines, thus minimizing hardware requirements.

3.5 Where to cache

Note in the previous patterns that data can be cached in different locations. Both the presentation server and application server (as implemented in WebSphere Application Server V4) support caching of static and dynamic pages. In addition, as described earlier, WebSphere Edge Server Caching Proxy can cache both dynamic and static pages.

The question that arises is this: where is the best place to cache data to satisfy future requests? In general, the recommendation is to cache pages as close to the client as possible. In our examples, this means that data should be cached in the DMZ as close to the protocol firewall as possible, minimizing the machines and network routes that must be passed through to retrieve the requested pages.

Part 2

Runtime patterns: guidelines

4

Technology guidelines

This chapter and the remaining chapters focus on high availability and high performance considerations that should be kept in mind in Runtime patterns. Specifically, software configurations within the DMZ will be discussed using WebSphere Edge Server to address high availability and high performance needs.

This chapter introduces WebSphere Edge Server, its main components that may be used to address high availability and high performance requirements in a Runtime pattern, and considerations to keep in mind when evaluating the benefits of each.

4.1 WebSphere Edge Server

IBM WebSphere Edge Server for Multiplatforms is Web infrastructure software that addresses the scalability, reliability and performance needs of e-business applications in both local and geographically distributed environments. Its functions incorporate leading-edge and robust caching and load balancing that together compensate for the inherent weakness of the Internet to support critical business applications and expectations. WebSphere Edge Server helps information technology administrators to provide better service to:

- External users who access documents and data stored on the enterprise's server machines
- Internal users who access the Internet

It helps you to host Web-accessible content and to provide Internet access more efficiently and economically. The name Edge Server indicates that the software usually runs on machines that are close (in a network configuration sense) to the boundary between an enterprise's intranet and the Internet.

4.1.1 Components

WebSphere Edge Server is made up of several main components which allow you to reduce Web server congestion, increase content availability and improve Web server performance:

1. Load Balancer

   The load balancing component, known as the Load Balancer, is a server that is able to dynamically monitor and balance back-end servers and applications in real time. It improves a Web site's availability, scalability and performance by transparently clustering edge, Web and application servers. The main advantage of the load balancing component is that it allows heavily accessed Web sites to increase capacity, since multiple back-end servers can be dynamically linked in a single entity that appears in the network as a single logical server.

   Note: In previous versions of WebSphere Edge Server, the Load Balancer component was referred to as Network Dispatcher. Throughout this book, references to Network Dispatcher refer to the Load Balancer component.

2. Caching Proxy

   The caching and filtering component, known as the Caching Proxy, is a server that provides highly scalable caching and filtering functions used in receiving requests and serving URLs. Since tunable caching is capable of supporting high cache hit rates, this component can reduce bandwidth costs and provide more consistent rapid customer response times. Additionally, the Caching Proxy is programmable via plug-ins, and it caches and invalidates dynamic content generated by the IBM WebSphere Application Server's dynamic cache.

   Note: In previous versions of WebSphere Edge Server, the Caching Proxy component was referred to as Web Traffic Express.

3. Edge Services architecture

   WebSphere Edge Services Architecture is an extension to the IBM WebSphere software programming model which brings the WebSphere software platform to the edge of the network, enabling performance, scalability and availability for sophisticated e-business applications. It distributes application processing capability to the edge platform as part of an end-to-end application framework. WebSphere Edge Services Architecture also addresses reliability, availability, security, manageability and quality of service, to provide a complete system to help ensure 24/7 uninterrupted operations. The major components of the WebSphere Edge Services architecture are the Caching Proxy and Load Balancer mentioned previously along with:

   a. Content Distribution - The Content Distribution Framework provides an infrastructure that can be used to distribute static, dynamic, and multimedia Web content, along with application components to production servers, rehosting servers and edge server caches throughout the network.
   b. Application Offload - A typical Web application setup consists of three tiers: Presentation, Business logic, and Data Store, co-located at the origin application server. Application Offload moves some of the Presentation and Business logic related processing to the edge of the network. It enables the Edge Server to perform page composition from cached and rehosted fragments and Web objects.

4.1.2 History

Previous versions of WebSphere Edge Server for Multiplatforms were called IBM WebSphere Performance Pack.

Previous versions of the Load Balancer component were referred to as Network Dispatcher or IBM Secureway Network Dispatcher.

Previous versions of the Caching Proxy component were referred to as Web Traffic Express.

4.2 Load Balancer

The Load Balancer component of WebSphere Edge Server acts as the entry point to the content servers resources. It can be used to balance the load on back-end servers within a local area network or a wide area network using a number of weights and measurements that are set dynamically. These back-end servers can be either entry points to the Internet (points-of-presence machines) or internal network servers.

The Load Balancer intercepts data requests from clients and forwards the requests to the back-end server machine that is currently best able to fill the request. It balances the load of incoming requests among a defined set of back-end machines, called a cluster, that service the same types of requests. It can distribute requests to many types of servers, including both HTTP origin servers and caching proxy machines. This load balancing is transparent to clients and other applications.

The Load Balancer consists of five components that can be used separately or together:

Dispatcher	Used to balance the load on back-end servers within a local area network or wide area network using a number of weights and measurements that are dynamically set by the Dispatcher. It does not use a domain name server to map domain names to IP address. MAC, NAT and NAPT forwarding are all part of Dispatcher.
Content Based Routing	Used with the Caching Proxy to load balance based on the content of the client request
Mailbox Locator	Used with the IMAP or POP3 protocols to choose an appropriate back-end server based on the user ID and password provided by the client.

Site Selector	Used to balance the load on servers within a local or wide area network using a DNS round-robin approach or a more advanced user-specified approach.
Cisco Consultant	Used to generate back-end server weighting metrics that are then sent to the Cisco CSS Switch for optimal server selection.

The following figure illustrates a typical Dispatcher configuration and the data flow from a client through the Dispatcher to a back-end server that is part of a cluster of servers defined to the Dispatcher machine.

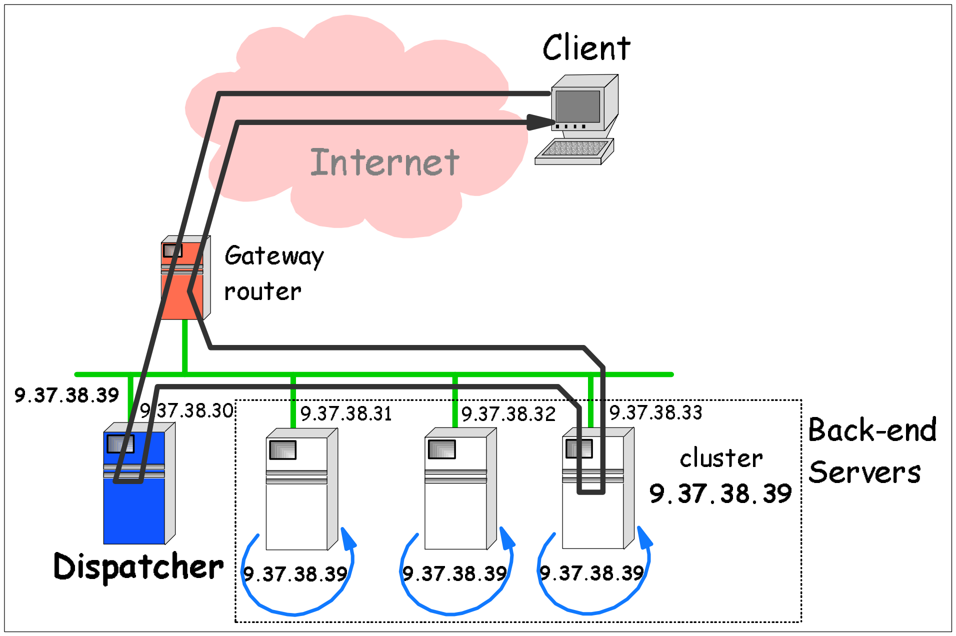


Figure 4-1 Typical dispatcher configuration

Note in Figure 4-1 that the data flow from the client to the back-end server passes through the Dispatcher component of the Load Balancer where the decision is made as to which back-end server is best suited to handle the client request. However, the response from the back-end server is sent directly to the client without having to pass through the Load Balancer machine. The actual TCP/IP packet is not modified by the Dispatcher. Load balancing is performed on machines within a cluster defined to the Dispatcher. The Dispatcher and all the back-end servers in the cluster have the same cluster IP address defined, but only the Dispatcher machine advertises the address to the Internet. Figure 4-1 is an example of Dispatcher MAC forwarding. NAT and NAPT forwarding are discussed later in this chapter.

The following figure is used to illustrate the main Dispatcher components.

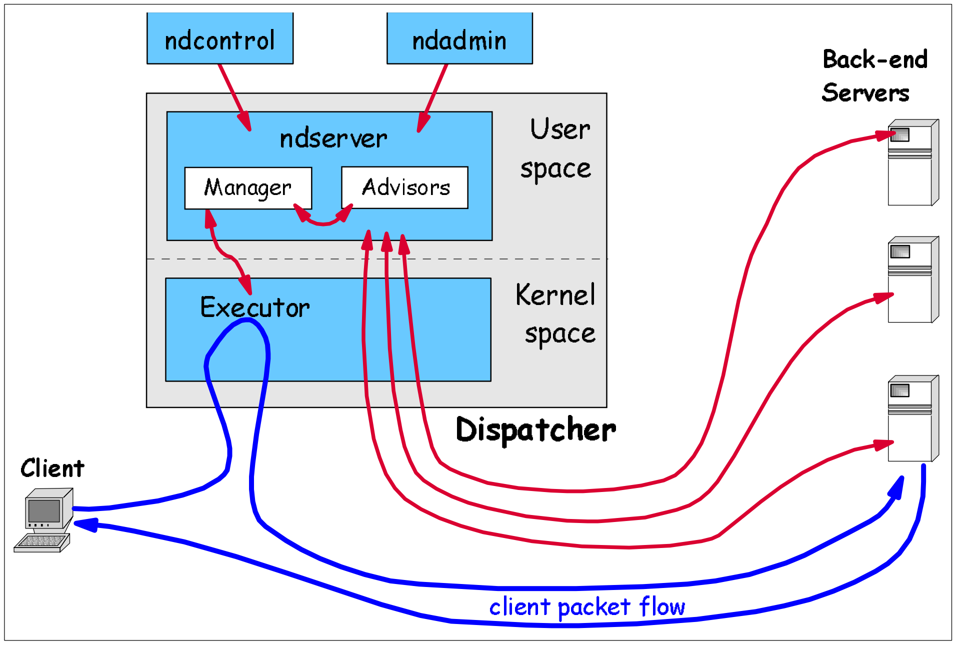


Figure 4-2 Dispatcher components - MAC forwarding example

The main Dispatcher components are as follows:

- Executor - The core function of the Dispatcher is the Executor. It is a kernel-level function (or device driver) which examines only the header of each packet and decides whether the packet belongs to an existing connection or represents a new connection request. The Executor uses a simple connection table stored in memory to achieve this. Note that the connection is never actually set up on the Dispatcher machine. The connection is between the client and the server, just as it would be if the Dispatcher were not installed. But the connection table records its existence and the address of the server to which the connection was sent. The Executor selects which back-end server to forward a new connection packet to based on information provided to it by the Manager.
- Manager - The Manager periodically sets the weights used by the Executor to determine the right back-end server to forward a new connection request to. The Manager can use four metrics to set the weights for back-end servers:
  - The count of active connections for each back-end server on each port. This count is maintained in the Executor.
  - The count of new connections for each server on each port since the last manager interval started. This count is maintained in the Executor.

  - A check that each back-end server has “displaceable capacity” which means that it can realistically process the work.
  - Feedback from Advisors that provide an application-level check that each server is up and running.
- Advisors - An Advisor is a lightweight client that runs as part of Dispatcher, but actually passes real commands to each back-end server. It only executes a trivial command. If this request is successful, the server is deemed to be up. If it fails, the Advisor informs the Manager, and the Manager sets the weight of that back-end server to zero until the server responds again.
- ndserver - A Java application that contains the user space threads in a Dispatcher configuration.
- cbrserver - A Java application that contains the user space threads in a CBR configuration.
- ndcontrol - A Java application to support command line configuration of the Load Balancer.
- cbrcontrol - A Java application to support command line configuration of the Content Based Routing component of the Load Balancer.
- ndadmin - A Java application to support a graphical user interface (GUI) configuration of the Load Balancer.

4.2.1 Load Balancer forwarding methods

The forwarding method can be selected when adding a port to a load balanced cluster of back-end servers. The default forwarding method value is media access control (MAC). You can specify the forwarding method parameter only when the port is added. Once the port has been added, you cannot change the forwarding method. Additional forwarding methods supported by the Load Balancer are Network Address Translation (NAT) and Network Address Port Translation (NAPT). Each is discussed in more detail below.

More detailed information about each forwarding method can be found in *IBM WebSphere Edge Server New Features and Functions in Version 2.0*; SG24-6511-00.

4.2.2 Media Access Control (MAC) forwarding

MAC forwarding is implemented in the Dispatcher component of Load Balancer. It permits load balancing of multiple back-end server machines running on the same subnet as the Dispatcher machine.

How it works

MAC forwarding load balances client requests across multiple back-end servers based on load calculations and input from advisors. Clients see one IP address for the entire site; this address is called the *cluster* address. The cluster address is aliased to the Dispatcher network interface card. Clients send requests to the cluster address. The firewall or router knows how to route to the Dispatcher server. In addition, the cluster address must be added to the loopback adapter on all the back-end servers that are part of the cluster. This must be done so that the back-end servers will accept unmodified packets forwarded to them from the Dispatcher machine.

The Dispatcher selects a back-end server to satisfy a client request, changes the destination MAC address, and forwards the *unmodified* data packet to the selected server. The back-end servers respond directly to the client without going back through the Dispatcher server. The Dispatcher keeps a record of each active session between clients and back-end servers so that additional data flows on the session will be sent to the same back-end server. When the session is ended, the record is removed from the Dispatcher machine (see Figure 4-3).

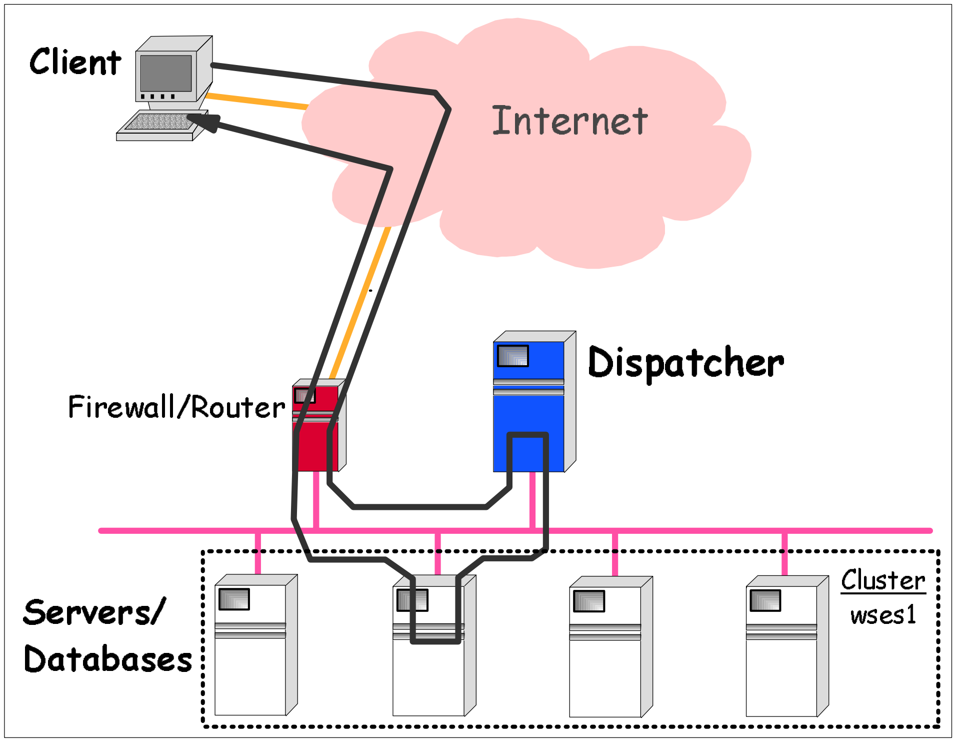


Figure 4-3 Dispatcher MAC forwarding

The Dispatcher MAC forwarding load balancing process is completely transparent to both the client and back-end server. The Dispatcher sees only the inbound client-to-server flows.

Considerations

The load balanced back-end servers must be configured with the cluster address on their loopback adapter.

The same data content must be available on each back-end server in a cluster.

Advantages:

- Requests account only for about 5% of the network traffic. Therefore, older server models are capable of balancing quite high volumes of requests.
- A separate high capacity return path may be configured to send requested data from the back-end servers directly to the client bypassing the Dispatcher subnet. This makes it more flexible to adapt to existing networks or changes to them.

Limitations:

- For MAC forwarding to work, the load balanced back-end servers must be on the same subnet at the Dispatcher machine. However, the back-end servers can be on different subnets from one another.

4.2.3 Network Address Translation (NAT) forwarding

NAT forwarding is implemented in the Dispatcher component of Load Balancer. It supports the following load balancing configurations:

- Multiple back-end server machines running on the same subnet as the Dispatcher machine.
- Multiple back-end server machines running on a different subnet from the Dispatcher machine.
- Multiple server daemons (each with their own IP address) running on the same back-end server machine.

How it works

NAT forwarding load balances client requests across multiple back-end servers based on load calculations and input from advisors. Clients see one IP address for the entire site: this address is called the *cluster* address. The Dispatcher advertises that address. The firewall or router knows how to route to the Dispatcher server. The Dispatcher selects a back-end server to satisfy a client request and forwards the *modified* packet to the selected server. The packet modifications involve the Dispatcher changing the source IP address from the client IP address to the *return* IP address of the Dispatcher machine, and changing the destination IP address from the cluster IP address to the IP address of the selected back-end server. The back-end servers respond to the client through Dispatcher server (using the `return` IP address) where the Dispatcher again modifies the source and destination IP addresses to their original values. See Figure 4-4.

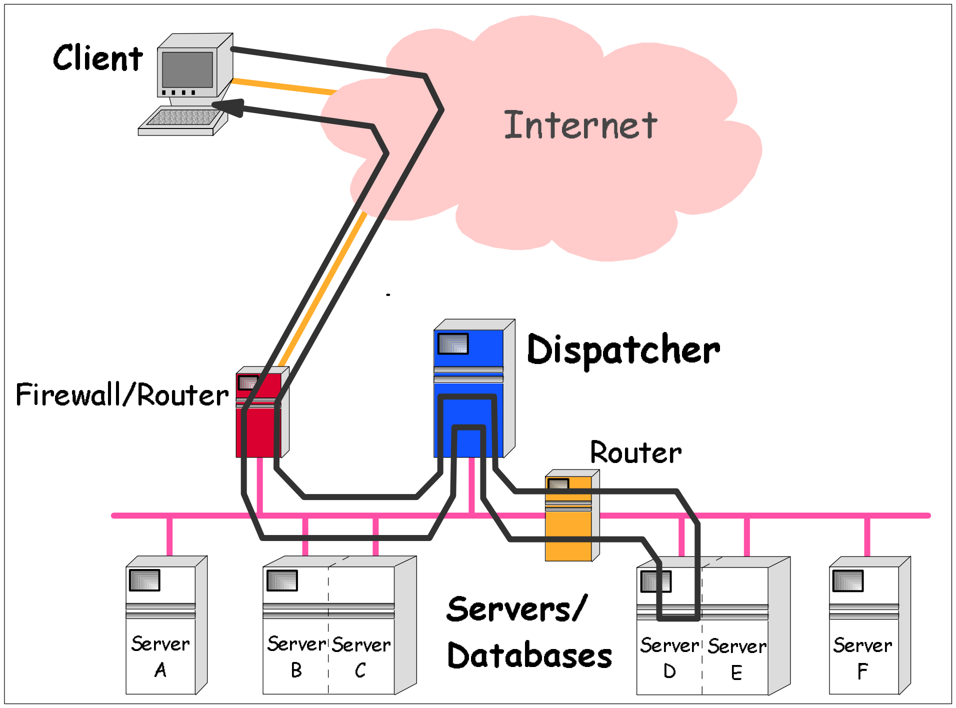


Figure 4-4 Dispatcher NAT forwarding to a different subnet

In Figure 4-4, the Dispatcher load balanced the client request to back-end Server D on a different subnet. Note that inbound and outbound flows must both go through the Dispatcher machine. If the Dispatcher had load balanced the client request to back-end Server C on the same subnet as the Dispatcher machine, inbound and outbound traffic flows would still go through the Dispatcher machine. Note that Servers D and E are two different server daemons running on the same back-end machine, each with its own IP address.

The Dispatcher NAT forwarding load balancing process is completely transparent to both the client and back-end server.

Considerations

The load balanced back-end servers do not require the cluster address to be configured on their loopback adapter.

NAT works well with upper-level application protocols such as HTTP, SSL, IMAP, POP3, NNTP, SMTP, Telnet, etc.

Advantages:

- NAT allows you to load balance content servers that are not on the local subnet.

Disadvantages:

- All response traffic is returned to the NAT server and the IP address is re-translated. This requires considerably more computing power than the MAC forwarding method

  Packet responses sent from the back-end servers to clients must return through the NAT server. Therefore, the Dispatcher machine must have bandwidth equal to the sum of all the back-end servers' bandwidth.
- The Load Balancer's implementation of NAT is a simple implementation of this feature. It only analyzes and operates upon the contents of TCP/IP packet headers. It does not analyze the contents of the data portion of the packets. For Load Balancer, NAT will not work with application protocols, such as FTP, which imbed the addresses or port numbers in the data portion of the messages. This is a well-known limitation of header-based NAT.

Note: If you do not set client gateway address to a non-zero value, the forwarding method can only be MAC.

4.2.4 Network Address Port Translation (NAPT) forwarding

NAPT forwarding is implemented in the Dispatcher component of Load Balancer. It supports the following load balancing configurations:

- Multiple back-end server machines running on the same subnet as the Dispatcher machine.
- Multiple back-end server machines running on a different subnet from the Dispatcher machine.
- Multiple server daemons running on the same back-end server machine with the same IP address; however, each server daemon listens on a different port.

How it works

NAPT forwarding load balances client requests across multiple back-end servers based on load calculations and input from advisors. Clients see one IP address for the entire site: this address is called the *cluster* address. The Dispatcher advertises that address. The firewall or router knows how to route to the Dispatcher server. The Dispatcher selects a back-end server to satisfy a client request and forwards the *modified* packet to the selected server. The packet modifications involve the Dispatcher changing the source IP address from the client IP address to the *return* IP address of the Dispatcher machine, and changing the destination IP address from the cluster IP address to the IP address of the selected back-end server. In addition, Dispatcher also modifies the destination port number to match the port on the selected back-end server. The back-end servers respond to the client through Dispatcher server (using the *return* IP address) where the Dispatcher again modifies the source and destination IP addresses to their original values (see Figure 4-5).

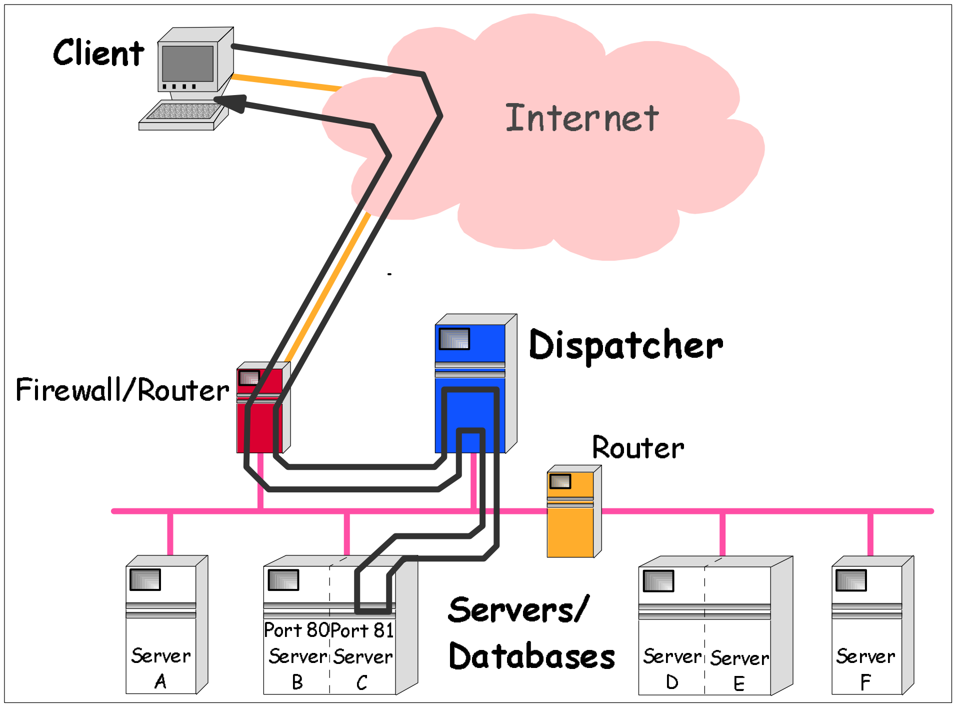


Figure 4-5 Dispatcher NAPT forwarding to the same subnet

In Figure 4-5, the Dispatcher load balanced the client request to back-end Server C on the same subnet as the Dispatcher. Note that inbound and outbound flows must both go through the Dispatcher machine. Note that Servers B and C are two different server daemons running on the same back-end machine with the same IP address, but different port numbers.

Considerations

The load balanced back-end servers do not require the cluster address to be configured on their loopback adapter.

The Load Balancer's Network Address Port Translation (NAPT) forwarding method works well with upper-level application protocols such as HTTP, SSL, IMAP, POP3, NNTP, SMTP, Telnet, etc.

Advantages:

- Allows you to configure multiple server daemons (running on the same physical server) to listen on different port numbers.
- Provides an additional layer of security by allowing a port change.

Disadvantages:

- All response traffic is returned to the NAPT server and the IP address is re-translated. This requires considerably more computing power than the MAC forwarding method.

  Packet responses sent from the back-end servers to clients must return through the NAPT server. Therefore, the Dispatcher machine must have bandwidth equal to the sum of all the back-end servers' bandwidth.
- The Load Balancer's implementation of NAPT is a simple implementation of this feature. It only analyzes and operates upon the contents of TCP/IP packet headers. It does not analyze the contents of the data portion of the packets. For Load Balancer, NAPT will not work with application protocols, such as FTP, which imbed the addresses or port numbers in the data portion of the messages. This is a well-known limitation of header-based NAPT.

4.2.5 Load Balancer High Availability feature

The Load Balancer High Availability feature is implemented in the Dispatcher component.

How it works

High availability involves the use of a second Dispatcher machine that monitors the main, or *primary*, machine and stands by to take over the task of load balancing should the primary machine fail at any time. See Figure 4-6.

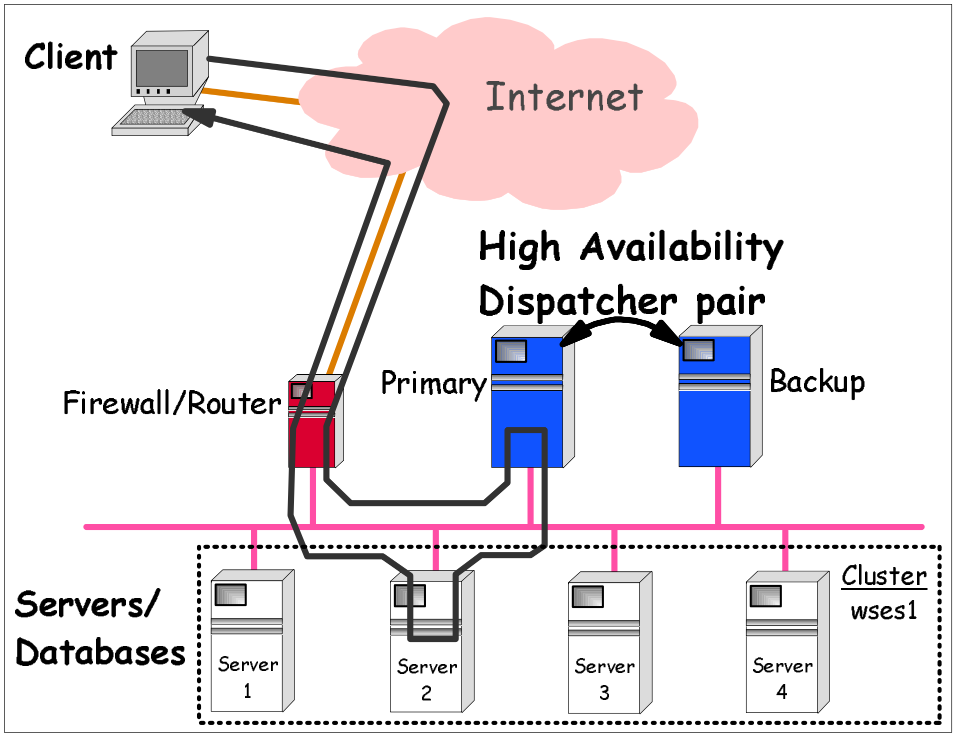


Figure 4-6 Dispatcher High Availability

Each Dispatcher machine has a specific role: primary or backup. In addition, each Dispatcher is in one of two states: active or standby. The primary machine is normally active and routes packets based on load balancing algorithms. The backup machine is normally in standby state and does not route. While the primary is active (load balancing), the backup is in a standby state, continually updated and ready to take over, if necessary. The active machine continually informs the standby machine of all the connections it is routing.

The standby machine is ready to take over and preserve connections in case the primary machine fails. If the backup machine detects that the active machine has failed, it will take over and begin load balancing. At that point, the statuses of the two machines are reversed: the backup machine becomes active and the primary machine becomes standby.

A “heartbeat” mechanism between the two Dispatcher machines is used to detect a Dispatcher failure. The standby machine makes the decision to take over if it detects that all heartbeats have stopped for two seconds. The standby machine then changes state to active. It also broadcasts gratuitous Address Resolution Protocol (ARP) commands so that everyone on the subnet (including the router) will now send packets for the cluster addresses to the standby (now active) machine.

Considerations

High availability is supported in MAC, NAT and NAPT forwarding configurations.

Both Dispatcher machines must be on the same LAN segment. In addition, both Dispatcher machines must have connectivity to the same clients and the same cluster of back-end servers.

When a standby Dispatcher machine takes over after a failure of the primary machine, most (but not all) of the existing in-flight connections are preserved. Clients see no disruption. Those connections that fail are typically connections that have just started or just ended.

The two Dispatcher configurations should be identical.

4.2.6 Load Balancer Mutual High Availability feature

The Mutual High Availability feature is implemented in the Dispatcher component of the Load Balancer.

How it works

Mutual high availability is similar to high availability in that it uses a second Dispatcher machine to monitor the main, or *primary*, machine. However, in the case of mutual high availability, the backup (standby) machine has clusters of its own that it is load balancing while acting as a standby machine for clusters on the other Dispatcher.

Both machines actively perform load balancing of a portion of the client traffic, and both machines provide backup for each other.

Each machine in a Dispatcher mutual high availability configuration has clusters defined to it for which it performs load balancing functions. Each machine performs the role of primary for the clusters it is load balancing, while at the same time acting as backup for clusters on the other Dispatcher machine. See Figure 4-7.

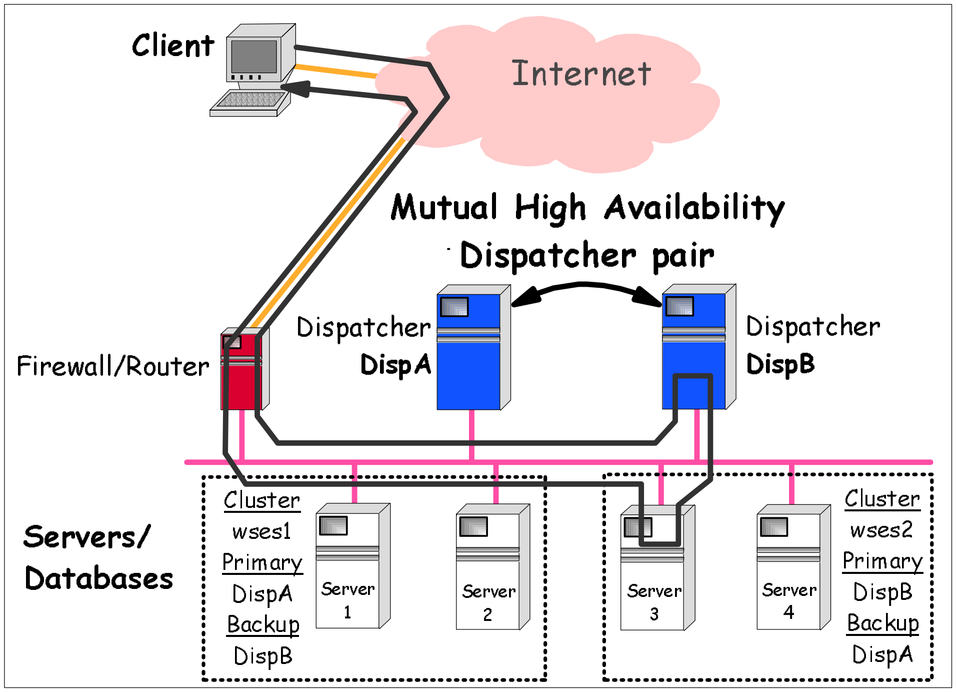


Figure 4-7 Dispatcher Mutual High Availability

In Figure 4-7, Dispatcher DispA is the primary machine for cluster wses1 and the backup machine for cluster wses2. Likewise, DispB is the primary machine for cluster wses2 and also the backup machine for cluster wses1. If either dispatcher machine fails, scripts on the other dispatcher machine are automatically executed to take over the load balancing responsibilities for the clusters on the failed machine.

The “heartbeat” mechanism described in 4.2.5, “Load Balancer High Availability feature” on page 71 also applies to a mutual high availability configuration.

Mutual high availability is different from high availability in that it is based specifically on cluster addresses rather than on a Dispatcher machine as a whole.

Considerations

Mutual high availability is supported in MAC, NAT and NAPT forwarding configurations.

Both Dispatcher machines must configure both their own clusters and the clusters they back up in the same way.

Both Dispatcher machines must be on the same LAN segment. In addition, both Dispatcher machines must have connectivity to the same clients and the same clusters of back-end servers.

When a standby Dispatcher machine takes over after a failure of the primary machine, most (but not all) of the existing in-flight connections are preserved. Clients see no disruption. Those connections that fail are typically connections that have just started or just ended.

4.2.7 Wide Area Network Dispatcher (WAND)

The Wide Area Network Dispatcher feature is implemented in the Dispatcher component of Load Balancer. It allows Dispatcher to load balance both local back-end servers and remote back-end servers in a wide area network environment.

How it works

Wide Area Network Dispatcher support involves the use of a local Dispatcher (with or without local back-end servers) and one or more remote Dispatchers, each with their own set of load balanced back-end servers. A remote Dispatcher and its associated back-end servers are referred to as a *remote server*.

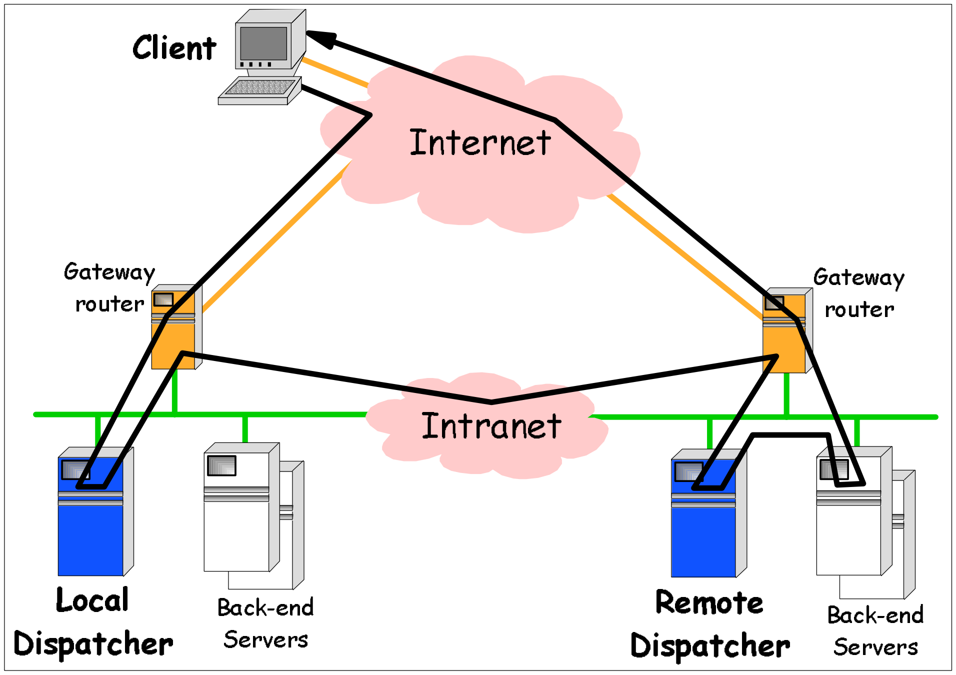


Figure 4-8 Wide Area Network Dispatcher support

The local Dispatcher continues to receive all inbound traffic for all servers, local and remote. A client's request flows from the Internet to the local Dispatcher machine. From there, it can be load balanced to one of the locally attached back-end servers, or sent to a geographically remote Dispatcher machine (remote server). The remote server load balances the request among its back-end servers, and the request is returned directly back to the Internet and client. This allows one cluster address to support all worldwide client requests while distributing the load to servers around the world. The local Dispatcher machine that initially receives the client request can still have local back-end servers attached to it and can distribute the load among its local servers as well as the remote servers.

The transmission of client requests from the local Dispatcher to a remote server is achieved by encapsulating the unmodified client packets at the local Dispatcher, and then un-encapsulating them at the remote Dispatcher. There are no packet modifications.

You can configure remote servers as far away as you like (with links preferably inside your private network, but also across the Internet) to provide a site which is geographically distributed over a few miles, or across the globe.

Considerations

Internet clients see only one IP address for their requests, just as before. Client packets to be balanced by Dispatcher go to the cluster address advertised by the local Dispatcher.

The local Dispatcher chooses the server (local back-end servers or remote servers). If a remote server is chosen, client data must be encapsulated by the local Dispatcher and un-encapsulated by the remote Dispatcher.

Data returned to the client from the back-end server uses the back-end server's gateway to go directly back to the client only if MAC forwarding (refer to 4.2.2, “Media Access Control (MAC) forwarding” on page 65) is used as the forwarding method on the remote Dispatcher cluster. The cluster address is the source address. Both local and remote Dispatchers see only the inbound client-to-server flows.

Avoid configuring the path to a remote Dispatcher back across the Internet. Use a private link through your intranet if possible.

4.2.8 Advisors

Advisors are lightweight clients that run inside the Dispatcher, providing information about the load and status of a given back-end server. Advisors periodically send requests to the back-end servers to measure actual response times for a particular protocol and determine the status of the server. If the transaction succeeds, the server is considered to be up and operational. The Load Balancer provides protocol-specific advisors for HTTP, FTP, Telnet, NNTP, IMAP, SMTP, SSL, ping, DB2, DNS, POP3, and others popular protocols.

To obtain the load value, the advisor:

- Pings and opens a connection with each back-end server
- Sends a protocol-specific request message

  The content of the message is specific to the protocol running on the server. For example, the HTTP advisor sends an HTTP “HEAD” request to the back-end server.
- Listens for a response from the back-end server
- Calculates the load value

  After getting the response, the advisor makes an assessment of the server. To calculate this “load” value, most advisors measure the time for the back-end server to respond, and then use this value (in milliseconds) as load.
- Reports the load value to the manager function.

  If the advisor determines that a back-end server is alive and well, it will report a positive, non-zero load number to the Manager. If a back-end server does not respond, the advisor returns a negative value (-1) for the load.

  The Manager calculates aggregate weight values from all its sources and sets these weight values into the executor function. The Executor then uses these weights for load balancing new incoming client connections. A back-end server that is down is given a weight of zero, and packets will not be forwarded to it until the server responds to the advisor again.

Refer to Figure 4-2 on page 64 for more detail on the executor and manager functions.

Custom advisors

Load Balancer supports writing your own advisors for specific applications. These are referred to as custom advisors and can be based on simple Java code provided with Load Balancer. Custom advisors execute on the Dispatcher node and must be written in the Java language.

4.2.9 Metric Server

The Metric Server function of Load Balancer provides back-end server load information to the Dispatcher manager component in the form of system-specific metrics (for example CPU utilization or percentage of memory used).

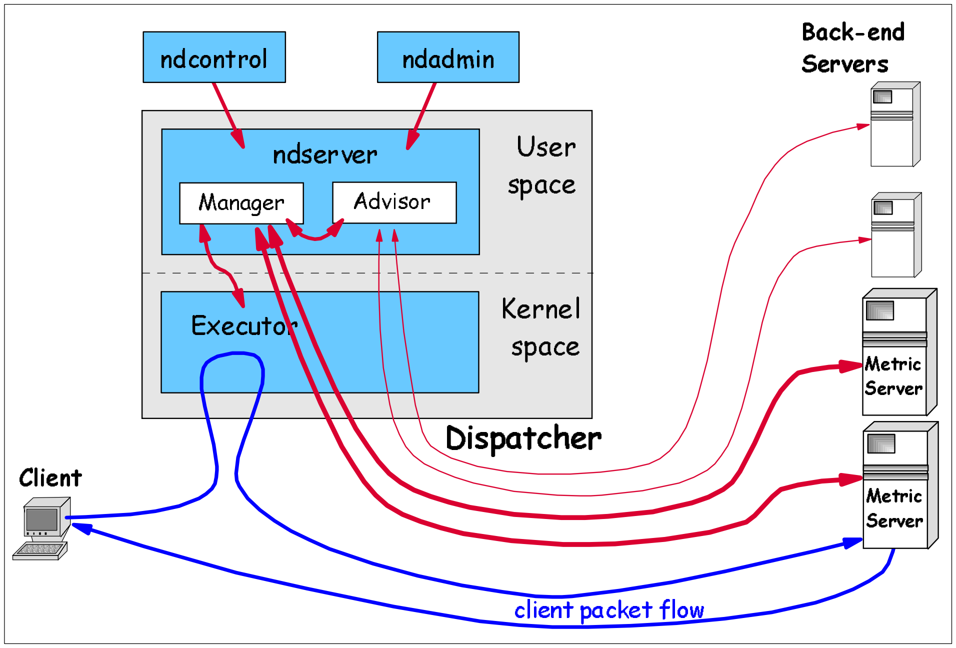


Figure 4-9 Metric server configuration

The Manager queries the Metric Server agent residing on the back-end servers and uses this information to assign weights for the load balancing process. The Metric Server agent must be installed and running on the back-end servers that are to be load balanced using Metric Server.

The Metric Server reports on back-end server systems as a whole, rather than on individual protocol-specific server processes.

4.3 Caching Proxy

The Caching Proxy component of WebSphere Edge Server is both a caching proxy server and a content filter. It can be used to provide a robust, efficient proxy server with an optional cache. The caching proxy server can be configured to operate as a forward proxy server for clients, as a transparent proxy server for clients or a reverse proxy server for other servers.

The Caching Proxy component acts as a gateway for multiple clients and performs basic Web server duties, such as receiving client requests and serving URLs. It intercepts data requests from a client, retrieves the requested information from content-hosting machines, and delivers that content back to the client. In addition, it can save or cache the Web documents it retrieves, and serve subsequent requests for those documents from its local cache. The client receives the requested information faster while network bandwidth and Web server processing time is reduced. Other features of caching are:

- The ability to handle very large caches
- An option to automatically refresh the cache with the most frequently accessed pages
- The ability to cache even those pages which the header information says to fetch every time
- Configurable daily garbage collection to improve server performance and ensure cache maintenance

The Caching Proxy allows you to set content filtering at the proxy server level, rather than or in addition to the browser level, where content filtering could be easily compromised or over-ridden.

The Caching Proxy consists of two components:

Caching Proxy	provides the proxy server and filtering function with the optional cache and cache management functions
TQoS (Linux only)	provides the Transaction Quality of Service plug-in for the Caching Proxy to allocate network bandwidth according to the class of client authentication

The following figure shows the Caching Proxy configured in the role of a reverse proxy.

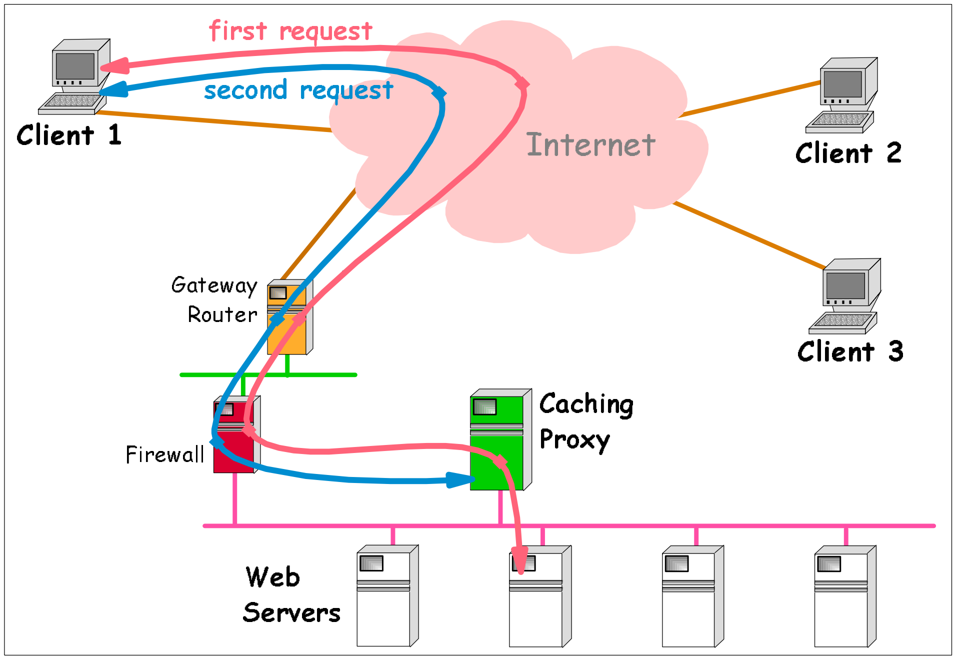


Figure 4-10 Reverse proxy configuration

Note in Figure 4-10 that Client 1 makes a request for data from a Web server. The first request for any data goes through the caching proxy where a cache search is performed. If the requested data is not in the cache, the request is forwarded to the appropriate origin Web server that can satisfy the request. The requested data is returned through the proxy server where it is cached and sent on to the original client (Client 1 in our example). The second request for the same data from any client (Client 1 in Figure 4-10) can be serviced by the proxy server's cache and does not go to the actual origin server. Requests for the same data from Clients 2 or 3 would also be serviced by the proxy server's cache.

4.3.1 Forward proxy

The Caching Proxy, when configured as a forward proxy, handles requests from multiple client browsers and retrieves data from the Internet for them. In addition, it can cache the returned pages to satisfy a request for the same pages at a later time. In this manner, a forward proxy can reduce the amount of traffic that traverses the Internet requesting duplicate data, while at the same time improving the response time for those requests.

How it works

A forward caching proxy server is typically placed between client machines in a branch office or ISP and the Internet. All client access to the Internet goes through the forward proxy. See Figure 4-11.

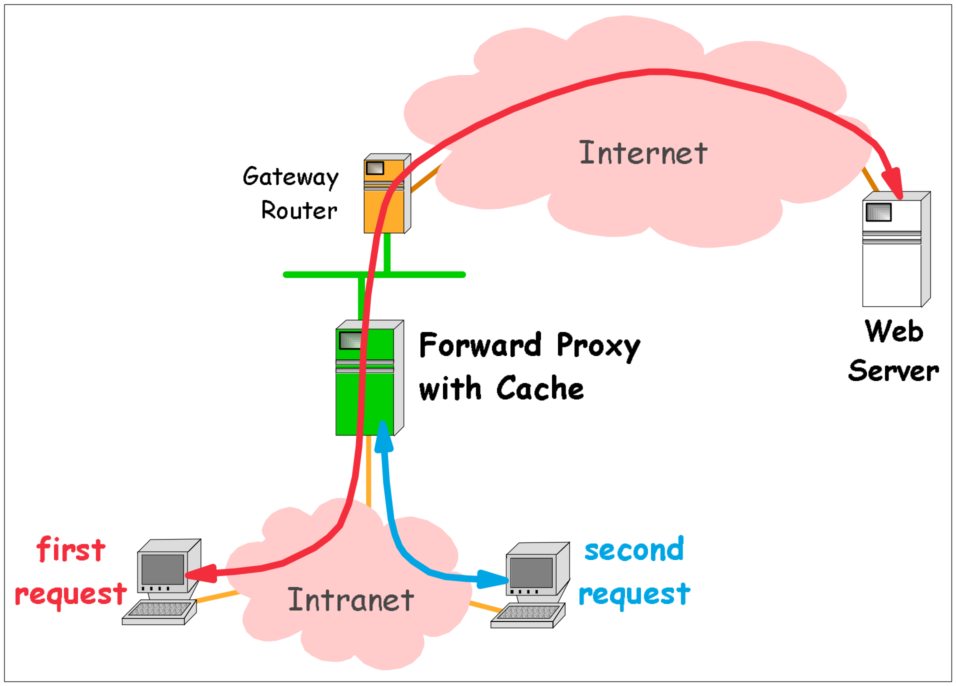


Figure 4-11 Forward Caching Proxy

A client's browser must be configured to direct requests to the forward caching proxy machine. When a client makes a request for a file or page stored on a Web server (*first request* in Figure 4-11), the forward proxy intercepts the request and checks its cache to determine if the requested file is already in its cache. If it is not, the forward proxy generates a new request with its own IP address as the originating address and sends the new request to the Web server.

The Web server then returns the requested information to the forward caching proxy instead of to the client. The forward proxy determines if the returned data is eligible for caching and if so, stores a copy in its cache. It then returns the requested data to the client.

Subsequent requests for the same data (*second request* in the figure) are satisfied by the data stored in the cache.

Cache management features allow you to determine which data is cached, establish how long it can remain in the cache, pre-fetch data that is highly requested, and so on.

Considerations

Individual client browsers must be configured to use the forward proxy server.

For the first request for a file or Web page, the forward (and reverse) caching proxy does not improve the efficiency of access to the Internet. The response for the first request is probably slower than without the caching proxy. Using a caching proxy yields benefits when other clients subsequently request the same files or Web pages.

A forward proxy can control:

- Who can get out to the Internet
- Where clients can surf to
- Where they can't

WebSphere Edge Serve Caching Proxy can proxy HTTP, FTP and Gopher protocols.

4.3.2 Reverse proxy

The Caching Proxy, when configured as a reverse proxy, acts on behalf of one or many back-end servers. A reverse Caching Proxy intercepts client requests arriving from the Internet, forwards them to the appropriate back-end server content hosts, caches the returned data (if requested) and delivers that data to clients across the Internet. The cached data can satisfy a request for the same pages at a later time. In this manner, a reverse proxy can reduce the amount of traffic and processing that a back-end server must perform to satisfy duplicate Internet requests for data, while at the same time improving the response time for those requests.

How it works

A reverse Caching Proxy server is typically placed between one or more back-end content servers and the Internet. It accepts requests from Internet clients for content stored on the reverse proxy server's home site. See Figure 4-12.

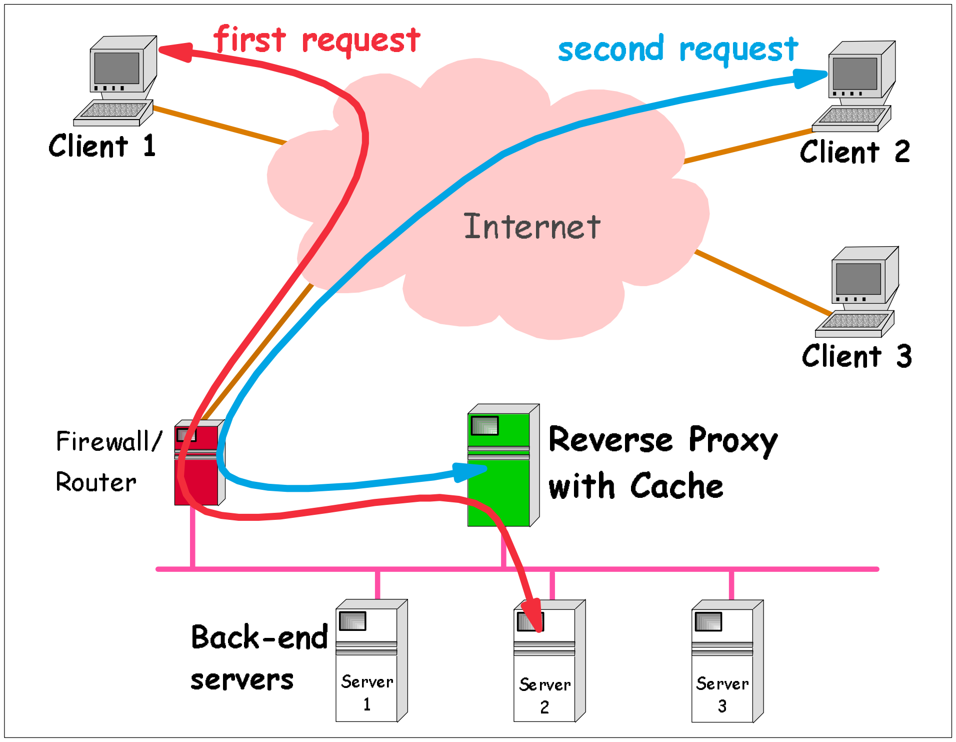


Figure 4-12 Reverse Caching Proxy

A reverse proxy server appears to the client to be the origin (content) server for requested data. The client is not aware that a request has been sent to another server. All client requests for data on the back-end servers pass through the reverse proxy machine. The proxy checks its cache to determine if the requested data is already cached. If it is (*second request* in Figure 4-12), then the requested data is returned directly to the client without involving the back-end server. If the requested data is not in the cache (*first request* in Figure 4-12), the reverse proxy generates a new request with its own IP address as the originating address and sends the new request to the back-end Web server.

The server then returns the requested information to the reverse caching proxy instead of to the client. The reverse proxy determines if the returned data is eligible for caching and if so, stores a copy in its cache. It then returns the requested data to the client.

As you can see when comparing the function and operation of a forward and reverse proxy, an important difference between the two is where the proxy machine is located within the network.

A reverse caching proxy can split the client request load among multiple back-end servers based on the URL requested by the client. For example requests for data from `http://www.lookhere.com/bells/...` could go to server 1 while requests for `http://www.lookhere.com/whistles/...` could go to server 2.

Considerations

It is not necessary to configure a client browser to use a reverse caching proxy server. The reverse proxy server presents itself to the Internet as the owner of the data requested from the back-end servers.

It is not necessary for the back-end servers to be on the same LAN segment as the reverse proxy server machine.

The proxy server can pre-fetch data from back-end Web servers that it or the administrator has chosen.

4.3.3 Cache

Static caching

A Web server that delivers a file must read it from disk and send it out to the network. Disk IO operations are among the most time consuming operations for a Web server. Saving such a file in memory and delivering it from there dramatically reduces the response time for all subsequent requests. This technique is known as *caching*. Today, most Web servers already use their own caches. Having redundant Web servers in a cluster results in redundant caching of files. Implementing a caching layer in front of the Web server clusters further improves the cache utilization and cache hit rate.

The Caching Proxy component of WebSphere Edge Server provides the base caching proxy server function. It intercepts requests from a client, retrieves the requested information from content-hosting machines, and delivers that information back to the client. In addition, it can store cachable content in a local cache (memory and/or disk) before delivering it to the requestor. Caching enables the Caching Proxy to satisfy subsequent requests for the same content by delivering it directly from the local cache, which is much quicker than retrieving it again from the content host.

Dynamic caching

For e-business applications, much of the memory and CPU processing power is used to generate the dynamic presentation of data. Caching the results of those operations is known as *dynamic caching*. It reduces response time to the client and reduces the load on both the redirector server nodes and the application server nodes. It also reduces the used bandwidth between the content server nodes.

Dynamic caching support can be configured separately on WebSphere Application Server V4. WebSphere Application Server can cache the dynamic output generated by a servlet/JSP and serve the subsequent requests from its cache.

With the WebSphere Edge Server Caching Proxy installed, it can also cache this type of dynamic Web content and serve subsequent requests from its cache thus reducing network bandwidth requirements and processing load on content servers. Dynamic caches on both WebSphere Application Server and the Caching Proxy server are synchronized when a new result is generated or a cached object expires.

4.4 Considerations for a high availability configuration

When designing a high availability configuration, you must weigh the possibility of an outage and the acceptable level of disruption to your system. For example, if you have an application that provides weather forecasts or telephone numbers, you may accept an outage of half an hour every couple of months. Another aspect is the likelihood of any failure at all. A Load Balancer node, for example, that is implemented on a stable Unix server and does a relatively simple task of load distribution is very unlikely to fail. On the other hand, an Application Server with half a dozen applications, connections to four different back-end systems, and an enormous amount of varying technologies on it, is very reasonably likely to be implemented in a high availability manner.

We have seen a few of the possibilities to consider when creating a highly available environment. These pattern variations focused on the existence of a load balancer and the possible configurations. The primary beneficiary of such a configuration is the user. But high availability does not only cover the client’s point of view. You as an owner will have additional requirements. A few examples:

- Maintenance without disruption of daily business
- Dynamic shifting of resources
- Constant development, testing and improvement parallel to production

These additional requirements will vary according to size, type of business, priorities, and so on.

Note: Increasing the number of content servers in your load balancer cluster also increases the load on the back-end systems. So the overall scalability will be limited by the capabilities of the back-end systems.

4.4.1 High availability configuration of nodes

In the high availability Runtime patters discussed in Chapter 2.1.1, “High availability rules for Runtime environments” on page 20, we focused on high availability of the Load Balancer. In reality, you will want to consider a similar configuration for every node. The possibilities will be determined by the capabilities of the products used to implement the nodes. Consider having backup nodes for your firewalls and server nodes.

4.4.2 Server affinity

Server affinity is a technique that enables the load balancer to remember which server was chosen for a certain client at his initial request. Subsequent requests are then directed to the same server again.This is important to maintain a session over multiple requests, because the HTTP protocol itself is stateless. WebSphere Edge Server load balancer supports server affinity (also known as *stickyness*) based on the client’s IP address.

When WebSphere Application Server receives an initial request from a client, it places a cookie in the IP header of the response it sends back to the client. This cookie will then be included in subsequent requests from this client. The WebSphere plug-in is able to recognize the cookie, which contains a unique ID (clone ID) of the Java Virtual Machine (JVM) that processed the initial request. This enables the plug-in to send the subsequent requests to the correct JVM (to both local and remote JVMs). In this case, stickiness should be turned off in load balancer in order to keep requests evenly distributed to the web server redirector nodes. The affinity handling is then left to the WebSphere plug-in.

4.4.3 Failover

Advanced techniques will even go one step further and provide failover in case of a problem occurring during an operation. Failover means that a user session can be recovered and continued on a different server without losing data that has been stored on the users behalf in the server’s memory. To enable failover, it is necessary to make a session persistent. To achieve this goal, the session data is written to a database node where it stored independent of a particular server in a session database. In case of a failing server, the request is redirected to another server and the session information is retrieved from the session database.

4.5 WebSphere plug-in

The WebSphere plug-in module is located on an HTTP Web server redirector node. This module is loaded by the HTTP server daemon into the kernel at startup time. The plug-in then forwards those requests that match URI rules to a defined application server group. These rules are in Extensible Markup Language (XML) format and can be automatically generated by WebSphere Application Server. The file is named plugin-cfg.xml and is located in the $WAS_HOME/config directory on the application server node. This file must be transferred to the HTTP Web server redirector node. By default, the plug-in expects to find this file in the $WAS_HOME/config directory (Yes, there will be a WebSphere directory tree on the Web server redirector node!).

Consulting the plugin-cfg.xml file, the plug-in decides which group of servers a request belongs to. Such a group of servers is referred to as a *server group* and these groups are identical to the server groups you can define on the application server node. The JVMs belonging to a server group are referred to as clones. A single application server (also a JVM) on the application server node results in a server group containing only the single JVM in the plugin-cfg.xml file.

The plug-in will also detect a failing JVM (a single JVM or a clone) within those server groups and mark it down. This JVM will not receive any requests until it recovered and got recognized by the plug-in as “back in business”.

5

Security guidelines

Security in the context of Web applications in general means protecting your resources and assets from theft, misuse or destruction of any kind.

More specifically, you should care about three things:

- Resource protection: make sure your security policy cannot be bypassed.
- Authentication: prove the identity of a user accessing the Web site.
- Authorization: grant certain rights to a user (person or machine), depending on its identity.

In order to keep performance at a high level, security matters can be deployed to specialized nodes instead of letting application server nodes do this in addition to their core business. Authentication proxy server nodes are of this special node type.

In this chapter, we present a brief overview of cryptography and security considerations, followed by a Runtime pattern to offload authentication and authorization processing from application server nodes.

5.1 The need for security

The Web and its associated technologies offer you virtually unlimited ways to extend the reach of your business information. Your critical business information and applications most likely reside on host systems in your private intranet. The quantity and quality of your business information, combined with the reach of the Web, afford you a unique opportunity to transform that information into a powerful competitive advantage. Merging Web technology with your existing information systems defines e-business, and building an e-business means finding a solution that provides a fast and cost-effective way to access, integrate and publish host information to Web-based clients. The Internet gives you an opportunity to extend the reach of your business to your employees, business partners or new customers around the world. However, you may have questions that make you reluctant to take advantage of the Internet's potential. The most talked-about concern over deploying applications over the Internet is security.

Network security is implemented to protect two objects:

- The data that is transmitted on the network
- The computers that are connected to the network

Network security cannot replace physical site security, host security on the connected systems, application security, and user security education. It can only act as a first line of defense. The network is replacing the physical doors of entry into your organization. Attackers from outside your organization must break through either your network or your physical security before they can attempt to break into your host systems. Security architecture can be defined as the following security services:

- Authorization

  This is the first thing that most of us think of when considering security. Authorization controls limit access to the facilities of a system to authorized users. Authorization is typically implemented on the target system and is always implemented in conjunction with the identification and authentication of the user.

- Identification and authentication

  Identification and authentication guarantee the identity of users and components of a system. Signed applet support provides identification of Java applets, while SSL supports client and server authentication.

- Confidentiality

  Confidentiality ensures that data cannot be accessed by unauthorized individuals. SSL is the facility most often used to provide this capability.

- Data integrity

  Data integrity protects against the unauthorized modification of data. This is another capability provided by SSL.

- Non-repudiation

  This service prevents the partners to a transaction from denying that data was sent or received. This is in many ways an extension to the identification and authentication and data integrity services. The technique of using digital signatures can be used to implement non-repudiation. On the Internet, the digital signature replaces the handwritten signature as a legal proof of authenticity. It may be used to provide proof of submission and proof of transport. Most Web servers and browsers today support digital certificates.

If you are concerned about information technology security in a wider context, you can find additional information at:

`http://www.ibm.com/security`

5.2 Concepts of cryptography and digital certificates

If you are sending data in the clear over a network that is not completely under your control from the receiver to the sender, you will be unable to ensure the following security functions:

- **Privacy**

  Anyone who is able to intercept your data might be able to read it.

- **Integrity**

  An intermediary might be able to alter your data.

- **Accountability or non-repudiation**

  It may be impossible to determine the originator of a message with confidence, and thus the person who sent the message could deny being the originator.

Security functions such as identification and authentication are also impacted because if authentication data such as passwords are sent without integrity and privacy, they can be intercepted in transit between sender and receiver, making the authentication compromised and worthless.

To ensure privacy, integrity and accountability in non-secure networks, cryptographic procedures need to be used. Today, two distinct classes of encryption algorithms are in use: symmetric and asymmetric algorithms. They are fundamentally different in *how* they work, and thus in *where* they are used.

5.2.1 Symmetric encryption algorithms

An encryption algorithm is called symmetric because the same key that is used to encrypt the data is also used to decrypt the data and recover the clear text (see Figure 5-1). The cipher and decipher processes are usually mathematically complex, nonlinear permutations.

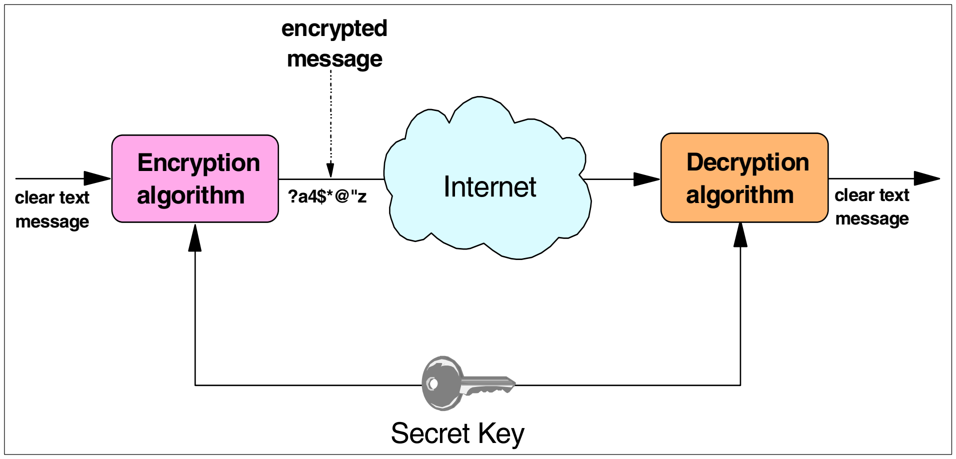


Figure 5-1 Symmetric encryption and decryption: using the same key

Symmetric algorithms are usually efficient in terms of processing power, so they are ideal for encryption of bulk data. However, they have one major drawback, which is key management. The sender and receiver on any secure connection must share the same key; in a large network where thousands of users may need to communicate securely, it is extremely difficult to manage the distribution of keys so as not to compromise the integrity of any one of them.

Frequently used symmetric algorithms include:

- **Data Encryption Standard (DES)**

  Developed in the 1970s by IBM scientists, Data Encryption Standard (DES) uses a 56-bit key. Stronger versions called Triple DES have been developed that use three operations in sequence: “2-key Triple DES” encrypts with key 1, decrypts with key 2, and encrypts again with key 1. The effective key length is 112 bits. “3-key Triple DES” encrypts with key 1, decrypts with key 2, and encrypts again with key 3. The effective key length is 168 bits.

- **Commercial Data Masking Facility (CDMF)**

  This is a version of the DES algorithm approved for use outside the U.S. and Canada (in times when export control was an issue). It uses 56-bit keys, but 16 bits of the key are known, so the effective key length is 40 bits.

- **RC2**

  Developed by Ron Rivest for RSA Data Security, Inc., RC2 is a block cipher with variable key lengths operating on 8-byte blocks. Key lengths of 40, 56, 64, and 128 bits are in use.

- **RC4**

  Developed by Ron Rivest for RSA Data Security, Inc., RC4 is a stream cipher operating on a bit stream. Key lengths of 40 bits, 56 bits, 64 bits, and 128 bits are in use. The RC4 algorithm always uses 128-bit keys; the shorter key lengths are achieved by “salting” the key with a known, non-secret random string.

- **Advanced Encryption Standard (AES)**

  As a result of a contest for a follow-on standard to DES held by the National Institute for Standards and Technology (NIST), the Rijndael algorithm was selected. This is a block cipher created by Joan Daemen and Vincent Rijmen with variable block length (up to 256 bits) and variable key length (up to 256 bits).

- **International Data Encryption Algorithm (IDEA)**

  IDEA was developed by James Massey and Xueija Lai at ETH in Zurich. It uses a 128-bit key and is faster than triple DES.

DES is probably the most scrutinized encryption algorithm in the world. Much work has been done to find ways to break DES, notably by Biham and Shamir, but also by others. However, a way to break DES with appreciably less effort than a brute-force attack (breaking the cipher by trying every possible key) has not been found.

Both RC2 and RC4 are proprietary, confidential algorithms that have never been published. They have been examined by a number of scientists under non-disclosure agreements.

With all the ciphers listed above, it can be assumed that a brute-force attack is the only means of breaking the cipher. Therefore, the work factor depends on the length of the key. If the key length is n bits, the work factor is proportional to 2**(n-1).

Today, a key length of 56 bits is generally only seen as sufficiently secure for applications that do not involve significant amounts of money or critically secret data. If specialized hardware is built (such as the machine built by John Gilmore and Paul Kocher for the Electronic Frontier Foundation), the time needed for a brute-force attack can be reduced to about 100 hours or less (see *Cracking DES:*

Secrets of Encryption Research, Wiretap Politics & Chip Design, by Electronic Frontier Foundation, John Gilmore (Editor), 1988). Key lengths of 112 bits and above are seen as unbreakable for many years to come, since the work factor rises exponentially with the size of the key.

5.2.2 Asymmetric encryption algorithms

Asymmetric encryption algorithms are so called because the key that is used to encrypt the data cannot be used to decrypt the data; a different key is needed to recover the clear text (see Figure 5-2). This key pair is called a public key and a private key. If the public key is used to encrypt the data, the private key must be used to recover the clear text. If data is encrypted with the private key, it can only be decrypted with the public key.

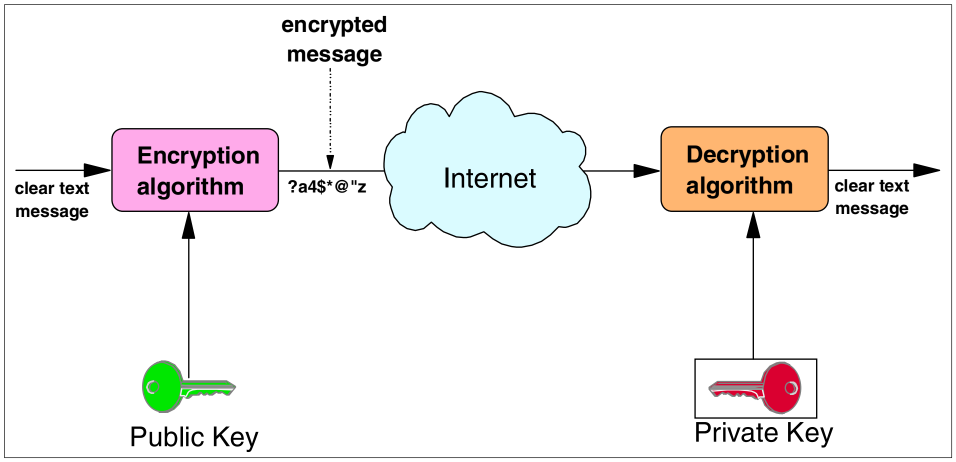


Figure 5-2 Public-key cryptography: using a key pair

Asymmetric encryption algorithms, commonly called Public Key Cryptography Standards (PKCS), are based on mathematical algorithms. The basic idea is to find a mathematical problem that is very hard to solve. The algorithm in most widespread use today is RSA. However, some companies have begun to implement public-key cryptosystems based on elliptic curve algorithms. With the growing proliferation of IPSec, the Diffie-Hellman algorithm is gaining popularity.

A brief overview of all three methods follows:

- **RSA**

  Invented 1977 by Rivest, Shamir, and Adleman (who formed RSA Data Security Inc.). The idea behind RSA is that integer factorization of very large numbers is extremely hard to do. Key lengths of public and private keys are typically 512 bits, 768 bits, 1024 bits, or 2048 bits. The work factor for RSA with respect to key length is sub-exponential, which means the effort does not rise exponentially with the number of key bits. It is roughly 2**(0.3*n).

- **Elliptic Curve**

  Public-key cryptosystems based on elliptic curves use a variation of the mathematical problem of finding discrete logarithms. It has been stated that an elliptic curve cryptosystem implemented over a 160-bit field has roughly the same resistance to attack as RSA with a 1024-bit key length. Properly chosen elliptic curve cryptosystems have an exponential work factor (which explains why the key length is so much smaller). Elliptic curve cryptosystems are now standardized by FIPS PUB 186-2, the digital signature standard (January 2000).

- **Diffie-Hellman**

  W. Diffie and M.E. Hellman, the inventors of public key cryptography, published this algorithm in 1976. The mathematical problem behind Diffie-Hellman is computing a discrete logarithm. Both parties have a public-private key pair each; they are collectively generating a key known only to them. Each party uses its own private key and the public key of the other party in the key generation process. Diffie-Hellman public keys are often called *shares*.

The beauty of asymmetric algorithms is that they are not subject to the key management issues that beset symmetric algorithms. Your public key is freely available to anyone, and if someone wants to send you a message he or she encrypts it using that key. Only you can understand the message, because only you have the private key. Asymmetric algorithms are also very useful for authentication. Anything that can be decrypted using your public key must have been encrypted using your private key, in other words, by you.

5.2.3 Performance issues of cryptosystems

Elliptic curve cryptosystems are said to have performance advantages over RSA in decryption and signing. While the possible differences in performance between the asymmetric algorithms are somewhere in the range of a factor of ten, the performance differential between symmetric and asymmetric cryptosystems is far more dramatic.

For instance, it takes about 1000 times as long to encrypt the same data with RSA (an asymmetric algorithm) as with DES (a symmetric algorithm), and implementing both algorithms in hardware does not change the odds in favor of RSA.

As a consequence of these performance issues, the encryption of bulk data is usually performed using a symmetric cryptosystem, while asymmetric cryptosystems are used for electronic signatures and in the exchange of key material for secret-key cryptosystems. With these applications, only relatively small amounts of data need to be encrypted and decrypted, and the performance issues are less important.

5.2.4 Cryptosystems for data integrity

Data integrity is the ability to assert that the data received over a communication link is identical to the data sent. Data integrity in an insecure network requires the use of cryptographic procedures. However, this does not imply that only the receiver is able to read the data, as with data privacy. Data could be compromised not only by an attacker, but also by transmission errors (although those are normally handled by transmission protocols such as TCP).

Message digest algorithms

A message digesting algorithm (often also called a “digital hash”) is an algorithm that “digests” (condenses) a block of data into a shorter string (usually 128 or 160 bits), which is called a message digest, secure hash, or Message Integrity Code (MIC). See Figure 5-3 for a graphical representation. The principle behind message digest algorithms is as follows:

The message cannot be recovered from the message digest.

It is very hard to construct a block of data that has the same message digest as another given block.

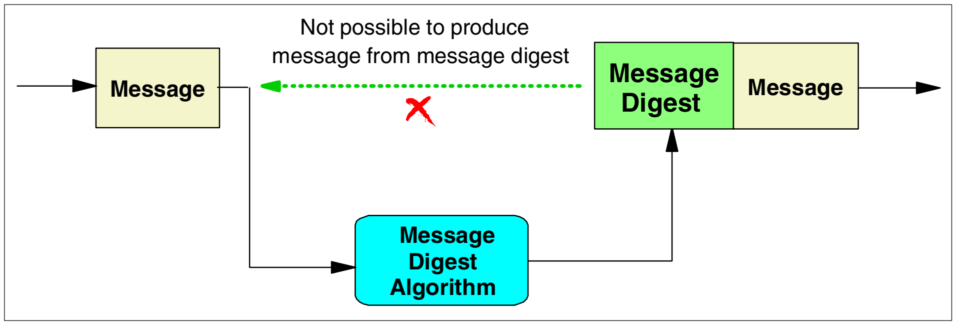


Figure 5-3 Message digest

Common message digest algorithms are:

- **MD2**

  Developed by Ron Rivest of RSA Data Security, Inc. The algorithm is mostly used for Privacy Enhanced Mail (PEM) certificates. MD2 is fully described in RFC 1319. Since weaknesses have been discovered in MD2, its use is discouraged.

- **MD5**

  Developed in 1991 by Ron Rivest. The algorithm takes as input a message of arbitrary length and produces as output a 128-bit message digest of the input. The MD5 message digest algorithm is specified in RFC 1321, *The MD5 Message-Digest Algorithm*. Collisions have been found in MD5; see *Cryptanalysis of MD5 Compress*, by Hans Dobbertin, available at:

  `http://www.cs.ucsd.edu/users/bsy/dobbertin.ps`.

- **SHA-1**

  Developed by the National Security Agency (NSA) of the U.S. Government. The algorithm takes as input a message of arbitrary length and produces as output a 160-bit “hash” of the input. SHA-1 is fully described in standard FIPS PUB 180-1, also called the Secure Hash Standard (SHS). SHA-1 is generally recognized as the strongest and most secure message digesting algorithm.

- **SHA-256, SHA-512**

  Developed by the National Security Agency (NSA) of the U.S. Government. The security of a hash algorithm against collision attacks is half the hash size and this value should correspond to the key size of encryption algorithms used in applications together with the message digest. Since SHA-1 only provides 80 bits of security against collision attacks, this is deemed inappropriate for the key lengths of up to 256 bits planned to be used with AES. Therefore, extensions to the Secure Hash Standard (SHS) have been developed. SHA-256 provides a hash size of 256 bits while SHA-512 provides a hash size of 512 bits.

Message digests for data integrity

The sender of a message (block of data) uses an algorithm, for example, SHA-1, to create a message digest from the message (see Figure 5-4). The message digest can be sent together with the message to provide data integrity. The receiver runs the same algorithm over the message and compares the resulting message digest to the one sent with the message. If both match, the message is unchanged.

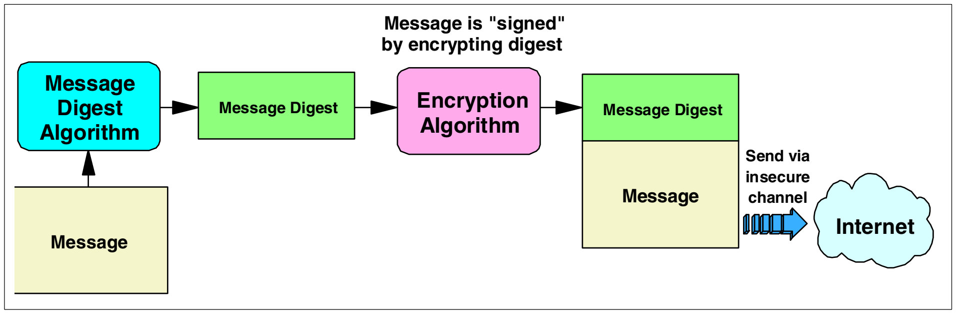


Figure 5-4 Message digest for data integrity

The message digest should not be sent in the clear: Since the digest algorithms are well-known and no key is involved, a man-in-the-middle could not only forge the message but also replace the message digest with that of the forged message. This would make it impossible for the receiver to detect the forgery. The solution for this is to encrypt the message digest, that is, to use a message authentication code (MAC).

Message authentication codes

Secret-key cryptographic algorithms, such as DES, can be used for encryption with message digests. A disadvantage is that, as in secret-key cryptosystems, the keys must be shared by sender and receiver. Furthermore, since the receiver has the key that is used in MAC creation, this system does not offer a guarantee of non-repudiation. That is, it is theoretically possible for the receiver to forge a message and claim it was sent by the sender. Therefore, message authentication codes are usually based on public/private key encryption in order to provide for non-repudiation. This is discussed further in 5.2.5, “Digital signatures” on page 99.

Keyed hashing for message authentication (HMAC)

H. Krawczyk and R. Canetti of IBM Research and M. Bellare of UCSD invented a method to create a message authentication code called HMAC, which is defined in RFC 2104 as a proposed Internet standard. A simplified description of how to create the HMAC is as follows: The key and the data are concatenated and a message digest is created. The key and this message digest are again concatenated for better security, and another message digest is created, which is the HMAC.

HMAC can be used with any cryptographic hash function. Typically, either MD5 or SHA-1 are used. In the case of MD5, a key length of 128 bits is used (the block length of the hash algorithm). With SHA-1, 160-bit keys are used. Using HMAC actually improves the security of the underlying hash algorithm. For instance, some collisions (different texts that result in the same message digest) have been found in MD5. However, they cannot be exploited with HMAC. Therefore the weakness in MD5 does not affect the security of HMAC-MD5.

HMAC is now a PKCS#1 V.2 standard for RSA encryption (proposed by RSA Inc. after weaknesses were found in PKCS#1 applications). For further details, see `http://www.ietf.org/rfc.html`. HMAC is also used in the Transport Layer Security (TLS) Protocol, the successor to SSL.

Message authentication used with SSL

In the Secure Sockets Layer Protocol (SSL), a slightly different MAC algorithm has been implemented. The MAC write-secret and the sequence number of the message are concatenated with the data, and a message digest is created. The MAC write-secret and this message digest are again concatenated for better security, and another message digest is created, which is the MAC. Again, for the hash function, either MD5 or SHA-1 can be used. If compression is used, the text is compressed before the MAC is calculated.

5.2.5 Digital signatures

Digital signatures are an extension to data integrity. While data integrity only ensures that the data received is identical to the data sent, digital signatures go a step further: they provide non-repudiation. This means that the sender of a message (or the signer of a document) cannot deny authorship; this is similar to signatures on paper. As illustrated in Figure 5-5, the creator of a message or electronic document that is to be signed uses a message digesting algorithm such as MD5 or SHA-1 to create a message digest from the data. The message digest and some information that identifies the sender are then encrypted with an asymmetric algorithm using the sender's private key. This encrypted information is sent together with the data.

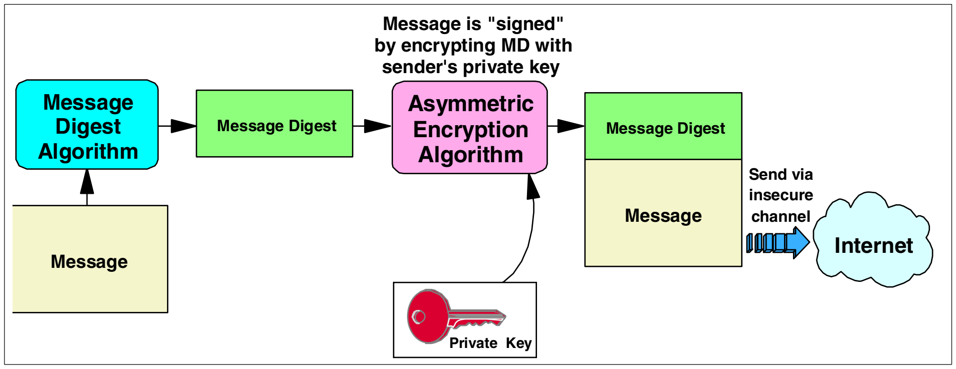

Message is "signed" by encrypting MD with sender's private key
Message Digest Algorithm
Message Digest
Asymmetric Encryption Algorithm
Message Digest
Message
Message
Send via insecure channel
Internet
Private Key

Figure 5-5 Digital signature creation

The receiver, as shown in Figure 5-6, uses the sender's public key to decrypt the message digest and identification of the sender. He or she will then use the message digesting algorithm to compute the message digest from the data. If this message digest is identical to the one recovered after decrypting the digital signature, the signature is recognized as valid proof of the authenticity of the message.

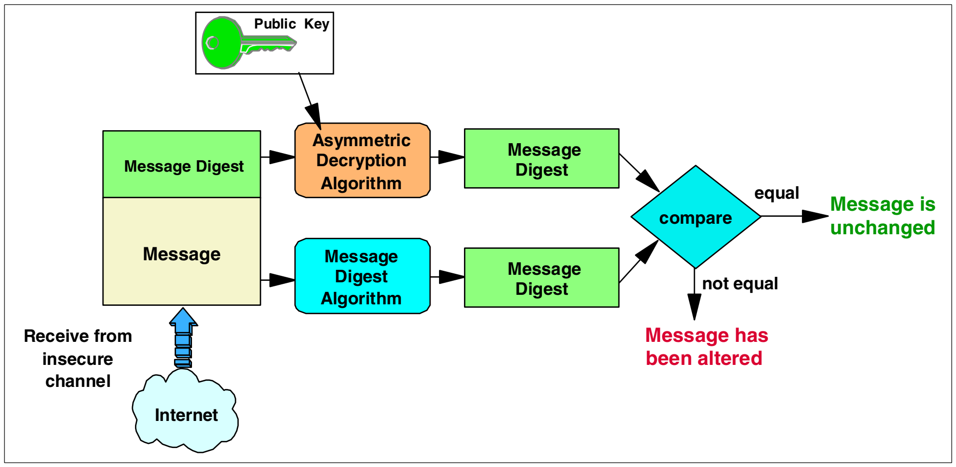


Figure 5-6 Digital signature verification

With digital signatures, only public-key cryptosystems can be used. If secret-key cryptosystems were to be used to encrypt the signature, it would be very difficult to make sure that the receiver (having the key to decrypt the signature) could not misuse this key to forge a signature of the sender. The private key of the sender is known to nobody else, so nobody is able to forge the sender's signature.

Note the difference between encryption using public-key cryptosystems and digital signatures:

- With encryption, the sender uses the receiver's public key to encrypt the data, and the receiver decrypts the data with his private key. This means everybody can send encrypted data to the receiver that only the receiver can decrypt. See Figure 5-7 for a graphical representation.

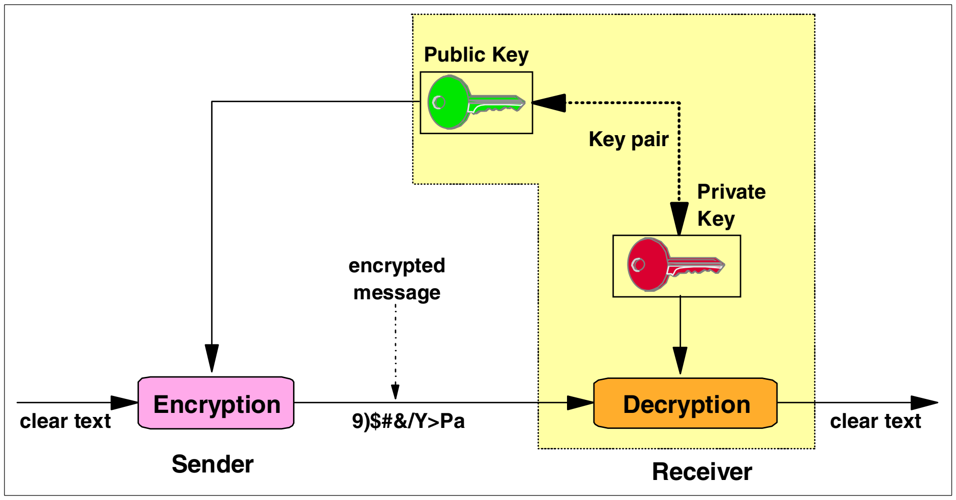


Figure 5-7 Encrypting data with the receiver's public key

- With digital signatures, the sender uses his private key to encrypt his signature, and the receiver decrypts the signature with the sender's public key. This means that only the sender can encrypt the signature, but everybody who receives the signature can decrypt and verify it.

The tricky part with digital signatures is the trustworthy distribution of public keys, since a genuine copy of the sender's public key is required by the receiver. A solution to this problem is provided by digital certificates, which are discussed next.

5.2.6 Public Key Infrastructure

A Public Key Infrastructure (PKI) offers the basis for practical usage of public key encryption. A PKI defines the rules and relationships for certificates and Certificate Authorities (CAs). It defines the fields that can or must be in a certificate, the requirements and constraints for a CA in issuing certificates, and how certificate revocation is handled.

PKI has been exploited in many applications or protocols, such as Secure Sockets Layer (SSL), Secure Multimedia Internet Mail Extensions (S/MIME), IP Security (IPSec), Secure Electronic Transactions (SET), and Pretty Good Privacy (PGP). PKI is described here, only insofar as its use with Web serving and Secure Sockets Layer (SSL) is concerned. For more information on PKI, refer to *Deploying a Public Key Infrastructure*, SG24-5512.

Digital certificates

When using a PKI, the user must be confident that the public key belongs to the correct remote person (or system) with which the digital signature mechanism is to be used. This confidence is obtained through the use of public key digital certificates. A digital certificate is analogous to a passport: the passport certifies the bearer's identity, address and citizenship. The concepts behind passports and other identification documents (for instance, drivers' licenses) are very similar to those that are used for digital certificates.

Passports are issued by a trusted authority, such as a government passport office. A passport will not be issued unless the person who requests it has proven their identity and citizenship to the authority. Specialized equipment is used in the creation of passports to make it very difficult to alter the information in it or to forge a passport altogether. Other authorities, for instance, the border police in other countries, can verify a passport's authenticity. If they trust the authority that issued the document, they implicitly trust the passport.

A digital certificate serves two purposes: it establishes the owner's identity and it makes the owner's public key available. Like a passport, a certificate must be issued by a trusted authority, the CA; and, like a passport, it is issued only for a limited time. When its expiration date has passed, it must be replaced.

Trust is a very important concept in passports, as well as in digital certificates. In the same way as, for instance, a passport issued by the governments of some countries, even if recognized to be authentic, will probably not be trusted by the US authorities, each organization or user has to determine whether a CA can be accepted as trustworthy.

For example, a company might want to issue digital certificates for its own employees from its own Certificate Authority; this could ensure that only authorized employees are issued certificates, as opposed to certificates being obtained from other sources such as a commercial entity such as VeriSign.

The information about the certificate owner's identity is stored in a format that follows RFC 2253 and the X.520 recommendation, for instance: CN=Heinrick Fienblatt O=IBM Corporation; the complete information is called the owner's distinguished name (DN). The owner's distinguished name and public key and the CA's distinguished name are digitally signed by the CA; that is, a message digest is calculated from the distinguished names and the public key. This message digest is encrypted with the private key of the CA.

Figure 5-8 shows the layout of a digital certificate.

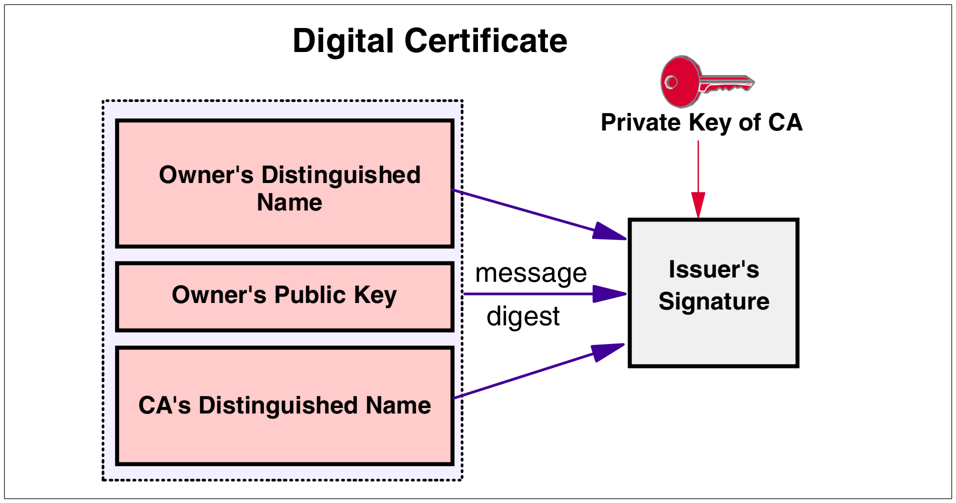


Figure 5-8 Simplified layout of a digital certificate

The digital signature of the CA serves the same purpose as the special measures taken for the security of passports such as laminating pages with plastic material; it allows others to verify the authenticity of the certificate. Using the public key of the CA, the message digest can be decrypted. The message digest can be recreated; if it is identical to the decrypted message digest, the certificate is authentic.

Security considerations for certificates

If I send my certificate with my public key in it to someone else, what keeps this person from misusing my certificate and posing as me? The answer is: my private key.

A certificate alone can never be proof of anyone’s identity. The certificate just allows the identity of the certificate owner to be verified by providing the public key that is needed to check the certificate owner’s digital signature. Therefore, the certificate owner must protect the private key that matches the public key in the certificate. If the private key is stolen, the thief can pose as the legitimate owner of the certificate. Without the private key, a certificate cannot be misused.

An application that authenticates the owner of a certificate cannot accept just the certificate. A message signed by the certificate owner should accompany the certificate. This message should use elements such as sequence numbers, time stamps, challenge-response protocols, or other data that allow the authenticating application to verify that the message is a “fresh” signature from the certificate owner and not a replayed message from an impostor.

Certificate Authorities and trust hierarchies

A user of a security service requiring knowledge of a public key generally needs to obtain and validate a certificate containing the required public key. To verify that the certificate is authentic, the receiver needs the public key of the CA that issued the certificate.

Most Web browsers come configured with the public keys of common CAs (such as VeriSign). However, if the user does not have the public key of the CA that signed the certificate, an additional certificate would be needed in order to obtain that public key. In general, a chain of multiple certificates may be required, comprising a certificate of the public key owner signed by a CA, and possibly additional certificates of CAs signed by other CAs. Many applications that send a subject's certificate to a receiver send not only just that certificate, but also all the CA certificates necessary to verify the certificate up to the root.

Obtaining and storing certificates

As has been discussed, certificates are issued by a CA. Clients usually request certificates by going to the CA’s Web site. After verifying the validity of the request, the CA sends back the certificate in an e-mail message or allows it to be downloaded.

Requesting server certificates

Server certificates can be either self-signed or they can be signed by an external CA. The server environment will determine which kind of certificate should be used: in an intranet environment, it is generally appropriate to use self-signed certificates. In an environment where external users are accessing the server over the Internet, it is usually advisable to acquire a server certificate from a well-known CA, because the steps needed to import a self-signed certificate might seem obscure, and most users will not have the ability to discern whether the action they are performing is of trivial consequence or not. It should also be noted that a root CA certificate received over a channel that is not trusted, such as the Internet, does not deserve any kind of trust.

5.3 Firewall concepts

A firewall machine is a computer used to separate a secure network from a non-secure network (Figure 5-9). Such networks are typically based on the TCP/IP protocol, but the concept of a firewall is not restricted to just TCP/IP.

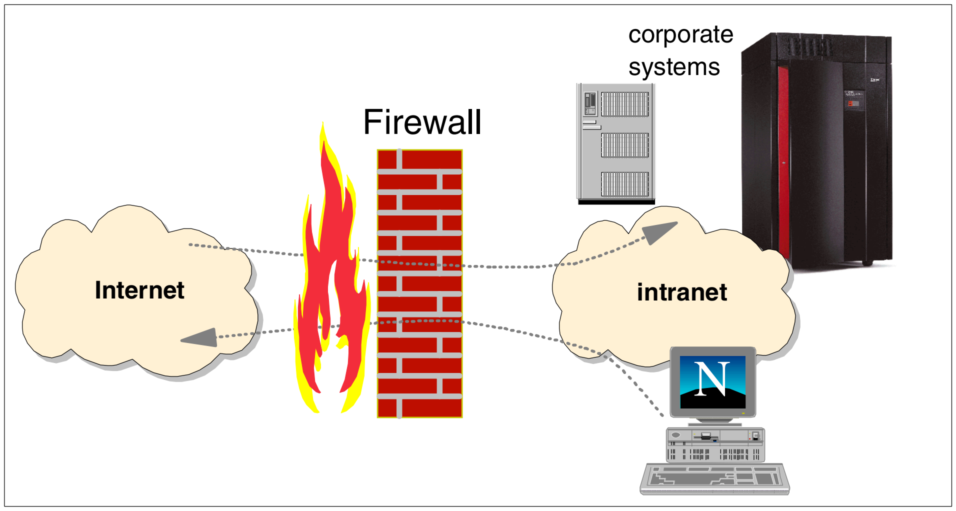


Figure 5-9 The firewall concept

Firewalls have become an important concept in TCP/IP-based networks, because the global Internet is a TCP/IP-based network and is often perceived as being a non-secure place to enter or traverse. Yet you still want your intranet (perceived as being a secure place) to be connected to the non-secure Internet.

The reasons for establishing connections between an intranet and the Internet are many, but generally fall into two categories:

- You want to provide a service to the Internet community or want to conduct business on the Internet.
- You want to allow your internal employees to access the vast amount of services on the Internet, as well as the ability to exchange or share information with other users on the Internet or through the Internet.

At this point, it might be useful to define the following terms:

- The term *intranet* refers to an internal TCP/IP network.
- The term *Internet* refers to the World Wide Web, and the associated infrastructure of news groups, e-mail, chat rooms and other services.
- The term *extranet* refers to TCP/IP networks of different companies connected with a secure connection, perhaps using virtual private network technology (VPN).

Doing e-business on the Internet is very different from just serving static information out of a Web server. Doing e-business means that you have to establish an environment where users on the Internet are able to interact with the applications and data that your daily existence as a company is based on and relies upon.

That data and those applications are likely, to a large extent, to be located in your environment, which means that you probably already are, or in the near-term future will be, challenged with the request to establish Internet access to your production environment.

When you connect your intranet to the Internet and define a strategy for how your firewall should function, you may think that it is sufficient to block all types of traffic that represent a risk, and allow the remaining traffic to pass through the firewall. However, such a strategy is based on the assumption that all risks are known in advance and that existing well-behaving traffic will remain well-behaving; such an assumption is a mistake. New ways of exploiting existing applications and well-known application protocols are being found every week, so an application that may be considered harmless today may be the instrument of an attack tomorrow.

5.3.1 General guidelines for implementing firewalls

A few general guidelines for implementing firewall technologies are worth including.

Before you start connecting your intranet to the Internet, define a security policy for how your firewall should function and how demilitarized zones should be configured. Decide what type of traffic is allowed through the firewall, and under what conditions, what kind of servers are to be placed in demilitarized zones, and what type of traffic is allowed between the demilitarized zone and the intranet.

When actually configuring your firewall, start by disallowing everything and then proceed by enabling those services you have defined in your security policy. Everything that is not specifically allowed should be prohibited.

If you establish more than a single gateway between your internal network and the Internet, make sure that all gateways implement the same level of security. It is common practice to use different firewall products in a vertical setup (product A between the Internet and the demilitarized zone and product B between the demilitarized zone and the intranet). That way, a hacker exploiting a vulnerability in product A is still stopped by product B. Of course, it does not make sense to use this concept in a horizontal setup (one gateway uses product A, the other one product B) because a hacker will get in at the weakest link.

If you build a perfect firewall on one end of your network while users on the other end dial in to the Internet from their LAN-attached PCs, enabling those PCs to act as IP routers between your internal network and the Internet, a hacker is soon going to exploit that back door into your network instead of wasting his time trying to break through your firewall.

One of the most important aspects of a firewall is its ability to log both successful and rejected access events. However, these logs are worth nothing if you do not set up daily administrative procedures to analyze and react to the information that can be derived from these logs.

By analyzing the firewall logs, you should be able to detect if unauthorized accesses were attempted and if your firewall protection succeeded in rejecting such attacks, or if it failed and allowed an intruder to gain access to resources that should not have been accessed. In addition, it might be a good idea to install an intrusion detection system.

This list is not all-inclusive, but merely points out some of the most important aspects of implementing firewall technologies in your network.

So far, the Internet has been considered to be the non-secure place, while your internal network has been considered the secure place. However, that may in some situations be an oversimplification. For example, consider a research department that works with highly confidential information. In such an environment, you may want to protect that research department from your regular users by implementing a firewall between your regular internal network and the network in your research department.

5.3.2 Firewall categories

There are many firewall technologies available, but they can in general be grouped into two major categories:

- Those that allow IP packets to be routed between two or more networks, namely packet-filtering routers.
- Those that disable IP routing, but relay data through specialized application programs, namely application-level gateways or proxies.

Packet filtering

A packet filtering router, as shown in Figure 5-10, is a special type of IP router. What differentiates a firewall packet filtering router from a normal IP router is that it applies one or more technologies to analyze the IP packets and decide if a packet is allowed to flow through the firewall or not. Such a firewall is sometimes also referred to as a screening filter, or router firewall.

Some packet-filtering techniques only act on data in the headers of individual packets, while others also look at data depending on the type of packet. The traditional packet-filtering router is stateless (each packet is handled independently) but there are products that save state over multiple packages and base their actions on the state information.

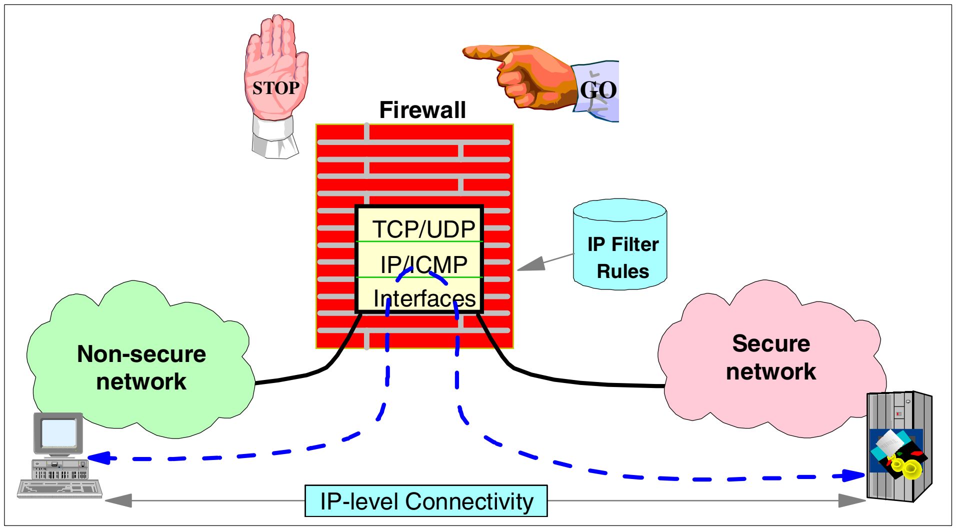


Figure 5-10 Packet filtering firewall

Application-level gateway

An application-level gateway, sometimes referred to as a *bastion host*, is a machine that disables IP-level routing between the non-secure network and the secure network, but allows specialized application gateway programs (termed *proxies*) that run on the firewall to communicate with both the secure network and the non-secure network (see Figure 5-11).

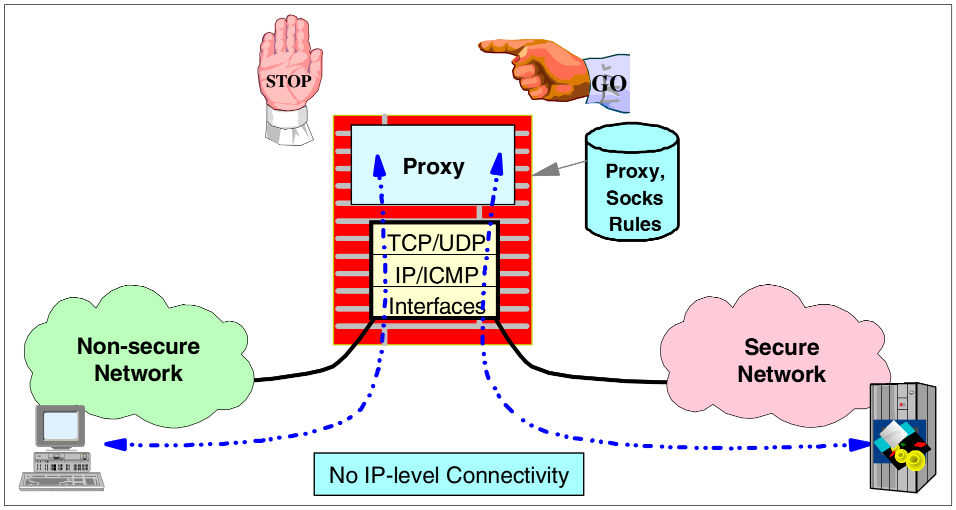


Figure 5-11 Application gateway firewall

The proxy applications on the firewall act as relay applications between users or applications on the secure and the non-secure networks. Examples of such proxy applications are HTTP or FTP proxy servers. The SOCKS server is also an application-level gateway, but a special kind, sometimes referred to as a circuit level gateway. A SOCKS server can relay all TCP and UDP connections, not just HTTP or FTP sessions. It does not provide any extra packet processing or filtering, and unlike proxy servers, it is often used for outbound connections through a firewall.

A firewall may not always have to be configured as either a packet-filtering router or as a proxy; it may be configured to perform the following functions:

- IP filtering
- Network address translation (NAT)
- Virtual private networks (VPN)
- FTP proxy server
- SOCKS server
- Domain name services

An excellent discussion of firewall technologies can be found in *TCP/IP Tutorial and Technical Overview*, GG24-3376.

The demilitarized zone

The demilitarized zone (DMZ) is a term often used when describing firewall configurations. Figure 5-12 shows a typical example. A DMZ is an isolated subnet between your secure network and the Internet. Much as the no-man's land between two entrenched armies, anyone can enter it, but the only elements present are those that you want to allow access to anyway. Nowadays, a demilitarized zone is an area in which you place the Web servers and other servers for public access, but which you also wish to protect to some degree.

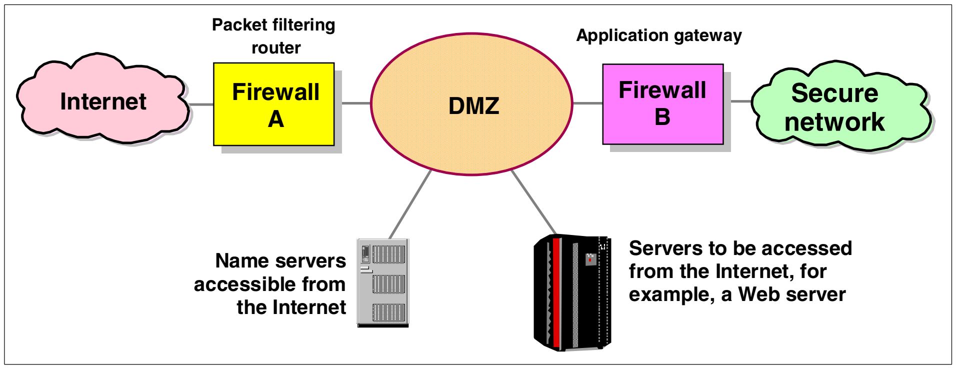

Packet filtering router
Application gateway
Internet
Firewall A
DMZ
Firewall B
Secure network
Name servers accessible from the Internet
Servers to be accessed from the Internet, for example, a Web server

Figure 5-12   A demilitarized zone

This is achieved by placing an outer firewall (often a packet-filtering router) between the Internet and the servers in the DMZ, and another firewall (often an application-level gateway) between your secure network and the DMZ. The outer firewall is designed to allow into the DMZ only those requests you wish to receive at your Web servers, but could also be configured to block denial-of-service attacks and to perform network address translation of the servers in your DMZ. The inner firewall is designed to prevent unauthorized access to your secure network from the DMZ and also perhaps to prevent unauthorized access from your secure network to the DMZ or the connected non-secure network.

When you put a server into a DMZ, it is strongly recommended that you use firewall technologies. You should use firewall technologies to block all traffic into and out of your server that does not belong to the services you are going to offer from this server. This control should be in place even if you already have a packet-filtering router or firewall between the insecure network and this server.

5.3.3 Hardening

Hardening is a process done to firewalls to make them more secure. All unnecessary services, user accounts, and software on the operating system are removed or disabled.

An operating system is designed to fit computers with different configurations performing different tasks. To accomplish this, extra items are installed with the operating system that will only be used in certain situations. Much of the time, these extra items just take up resources and might cause the computer to run more slowly. On a firewall, these items become more of a problem. They become unnecessary, and potential security exposures.

Almost everything on a firewall is a potential security exposure. By disabling and removing the unnecessary items, there will be less exposure for a hacker to exploit. All services, user IDs, and software on a firewall should be required only by the operating system or the firewall. Everything else should be removed.

Many firewalls will perform a limited amount of hardening. However, it is the responsibility of the firewall administrator to finish the task.

5.4 Virtual private network (VPN) and IPSec

A virtual private network (VPN) provides secure connections across the Internet, by establishing a “tunnel” between two secure networks. It is a generic solution that is application- and protocol-independent. A VPN encapsulates the IP datagram into another IP datagram in order to maintain data privacy. It can be used by two disparate parts of a corporation to connect their internal private networks by means of a non-secure network such as the Internet. An example of a VPN configuration is shown in Figure 5-13.

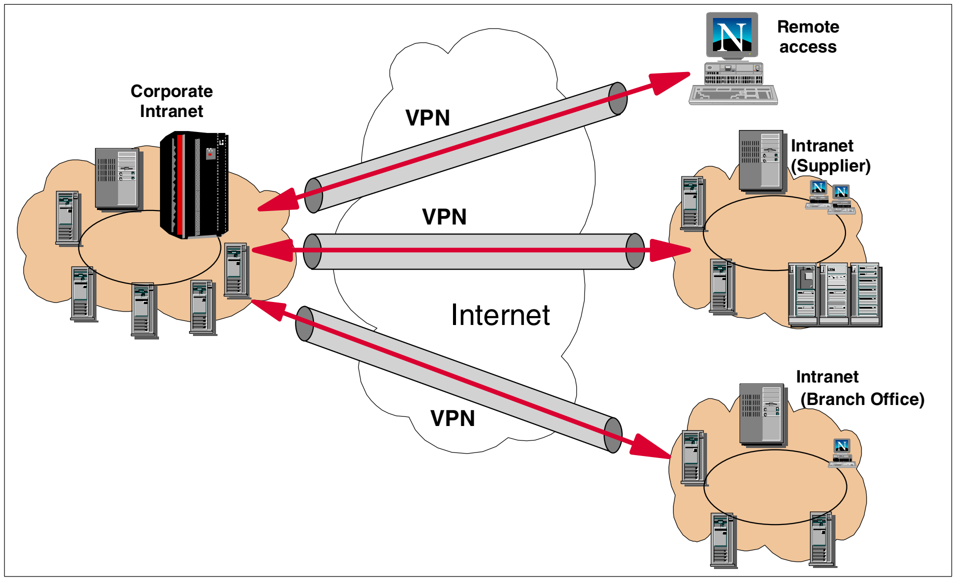


Figure 5-13 Virtual private networks

5.4.1 IPSec

In Figure 5-14, the TCP/IP layered protocol stack is shown, with the security-related protocols associated with each layer:

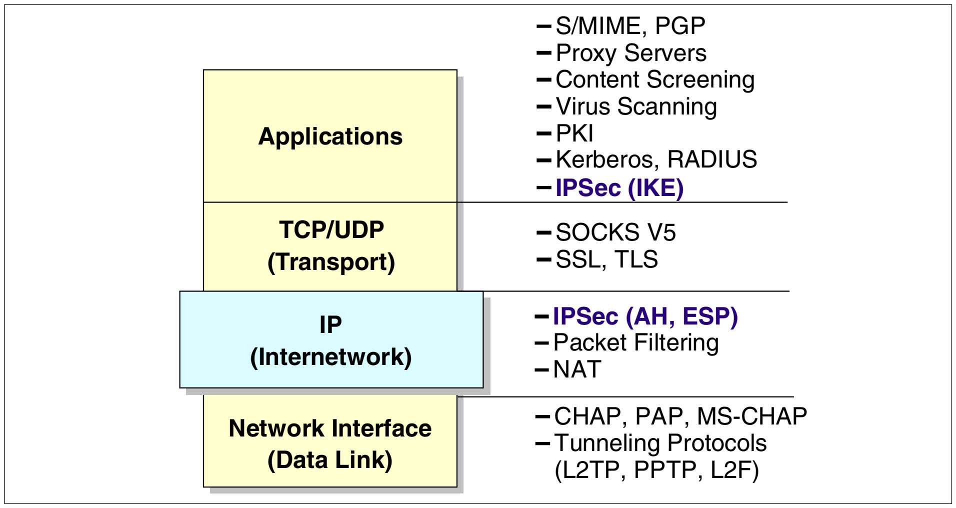


Figure 5-14 The TCP/IP protocol stack and the security-related protocols

Within the layered communications protocol stack model, the network layer (IP in the case of the TCP/IP stack) is the lowest layer that can provide end-to-end security. Network-layer security protocols provide blanket protection for all upper-layer application data carried in the payload of an IP datagram, without requiring a user to modify the applications.

The IP Security Architecture (IPSec) open framework is defined by the IPSec Working Group of the IETF. IPSec is called a framework because it provides a stable, long-lasting base for providing network layer security. It can accommodate today's cryptographic algorithms, and can also accommodate newer, more powerful algorithms as they become available. IPV6 implementations must support IPSec, and IPv4 implementations are strongly urged to do so.

IPSec is comprised of a number of components described in individual RFCs that are designed to operate together:

- Security Protocols - IP Authentication Header (AH) provides data origin authentication, data integrity, and replay protection while IP Encapsulating Security Payload (ESP) provides data confidentiality, data origin authentication, data integrity, and replay protection.
- Security Associations - an SA is a kind of session between two hosts defining the protocols to be used when transmitting data. ISAKMP (Internet Security Association and Key Management Protocol) is a generic framework for negotiating SAs and keys.
- Key Management - Internet Key Exchange (IKE) provides a method for automatically setting up security associations and managing and exchanging their cryptographic keys.

Security Associations

An IPSec Security Association (SA) corresponds to a session between two hosts. It defines the set of protocols and, with these, the negotiated algorithms and keys that are to be used when transmitting data between two hosts. An SA for data traffic is always unidirectional, so for a pair of hosts that are to communicate securely, at least two SAs, one for each direction, are needed. This differs from other protocols that make use of sessions such as, for instance, SSL. An SSL session covers the transmission in both directions.

Negotiating Security Associations (ISAKMP and IKE)

Before any data can be sent between two hosts using IPSec, a SA needs to be established. The IPSec architecture provides two methods for establishing an SA: a manual tunnel or ISAKMP/IKE.

With manual tunnels, the SA and keying material are generated on one of the hosts (the tunnel owner), transferred to the other host (the tunnel partner) with an out-band transport mechanism, and then imported. This procedure needs to be repeated whenever the validity of the keys has expired and new keying material needs to be generated.

Contrary to manual tunnels, ISAKMP and IKE provide automatic management of sessions and keys. ISAKMP provides a generic framework for the negotiation of SAs and keying material. It defines the procedures and packet formats to establish, negotiate, modify and delete SAs, but it does not provide any specific key-generation techniques or cryptographic algorithms.

Internet Key Exchange (IKE) is based on two protocols: Oakley (*The Oakley Key Determination Protocol*, by H. Orman; RFC 2412, November 1998) and SKEME (*SKEME: A Versatile Secure Key Exchange Mechanism for the Internet*, by H. Krawczyk; IEEE Proceedings, 1996). For the key exchange, Diffie-Hellman (DH) shares are used and the shared key thus obtained is used to derive the keys for data encryption and message authentication. Authentication can be performed with one of three alternatives:

- Digital signatures
- Public key encryption
- A shared secret (a key previously known to both parties)

The use of DH shares causes the connection to have a property called “perfect forward secrecy”. This means that even if the keys for one session are completely compromised, the keys for previous sessions are still safe.

Phases: it takes two

Two hosts can communicate with each other in many different ways that may need different sorts of protection. For instance, some traffic may need encryption and authentication, while other traffic may only need authentication.

IKE uses a two-phase approach to be able to meet these different needs with minimal overhead. In phase 1, an ISAKMP SA is negotiated to create a secure, authenticated channel between the two hosts. The ISAKMP SA is a single, bidirectional security association. In phase 2, the SAs for the individual type of traffic (one SA for each direction) are negotiated using the authenticated channel established in phase 1.

Due to the Diffie-Hellman key exchange and the authentication, phase 1 is computationally rather expensive. Phase 2 does not involve key exchange nor authentication and is much less expensive. Performing phase 1 just once for a pair of hosts and then multiple phase 2 operations for the individual connections is a concept that can improve performance considerably.

Identity protection

In phase 1, certificates and authentication data are exchanged between the hosts. IKE offers *identity protection*, meaning that all information that could identify a host to an attacker or eavesdropper can be encrypted. Depending on whether identity protection is really required, IKE supports two modes for phase 1: *main mode* offers identity protection, while *aggressive mode* does not. In main mode, a shared, secret key is established before the identification information (for instance, the host's digital certificate) is sent. For a diagram showing IKE main mode, see Figure 5-15.

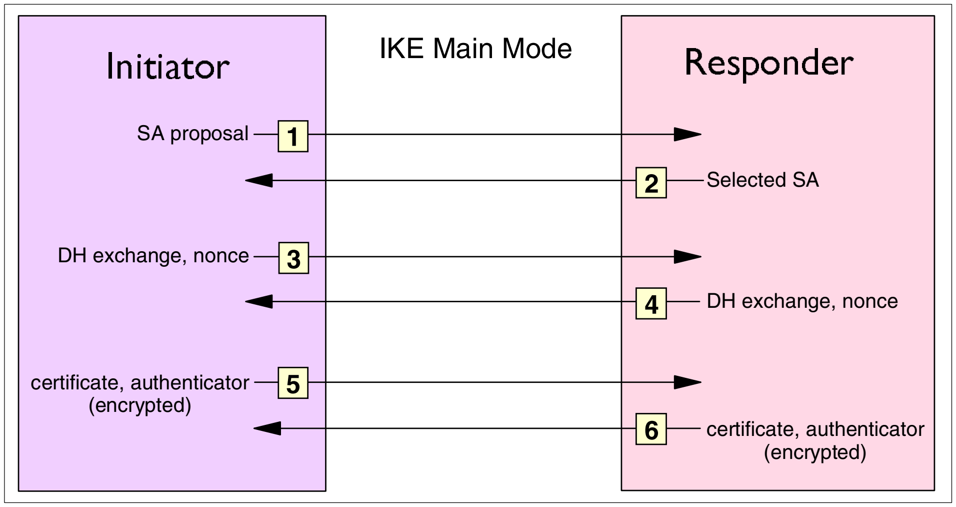


Figure 5-15 IKE phase 1 main mode

Aggressive mode does not require the DH key exchange to be completed before sending the remaining information. Therefore, there is only one exchange of messages in aggressive mode (see Figure 5-16).

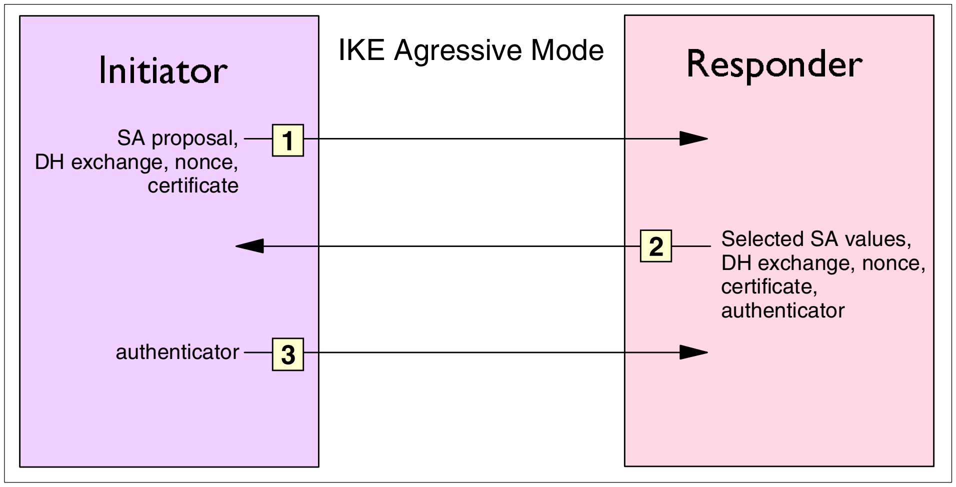

IKE Agressive Mode
Initiator
Responder
SA proposal, DH exchange, nonce, certificate
1
2
Selected SA values, DH exchange, nonce, certificate, authenticator
authenticator
3

Figure 5-16 IKE phase 1 aggressive mode

The exchange of messages taking part in phase 2 (negotiation of the SAs for the individual type of traffic) is called *quick mode*. In this mode, the pair of SAs for the intended type of communication is set up. The required keys for encryption and message authentication are generated from the shared key obtained in phase 1.

Transmitting data with IPSec

When a host wants to transmit one or more packets to another host it had not contacted before, it will perform the necessary IKE exchanges to set up the required SAs with the other hosts. Once this has all been performed and the necessary keys are generated, the host proceeds to send the first packet.

IPSec has two formats for sending data, which serve slightly different purposes. *Authentication Header* (AH) provides for message authentication and replay protection, whereas *Encapsulating Security Payload* (ESP) provides for data encryption in addition to message authentication and replay protection. The SA for a communication selects whether AH, ESP, or a combination of both is to be used.

Depending on the type of VPN connection between the two hosts, there are two modes, *tunnel mode* and *transport mode*, that are to be used.

ESP and AH in transport mode

If a VPN connection is being established between two hosts that are the endpoints for the packets transmitted between them, transport mode should be used. Figure 5-17 shows the format of an Authentication Header (AH) in transport mode.

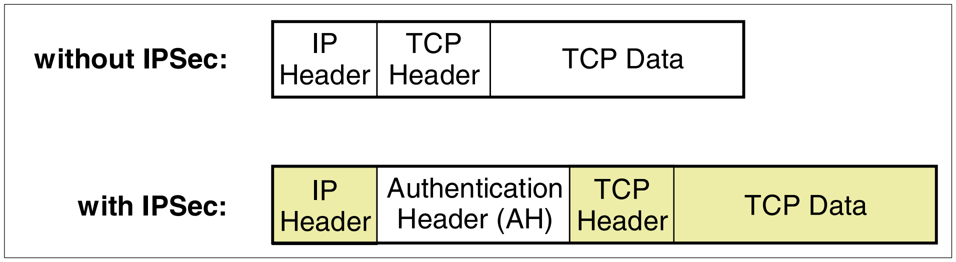


Figure 5-17 AH in transport mode

The message authentication applied by AH protects the parts of the packet that are shaded in Figure 5-17. Note that although the IP header is shaded in the diagram, parts of it are not authenticated because they can change in transit between sender and receiver.

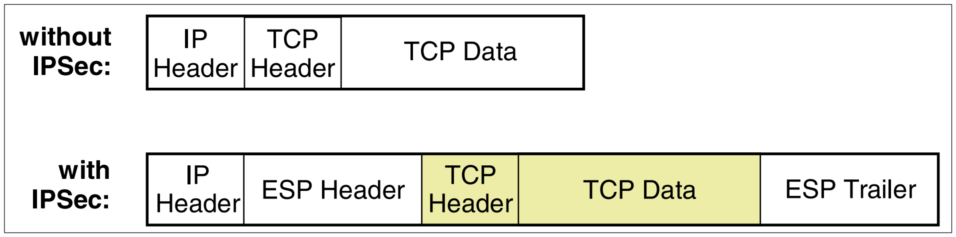


Figure 5-18 ESP in transport mode

With the Encapsulating Security Payload (ESP) format in transport mode, the TCP header and data are encrypted and, optionally, authenticated. But as can be seen in Figure 5-18 (the protected areas are shaded), the IP header is afforded no protection at all. However, this should not be a problem because sending and receiving hosts have been authenticated and verified in the SA.

ESP in tunnel mode

A common application of VPNs is the use of a protected tunnel between two secure networks. IPSec-capable firewalls at each end of the tunnel encrypt the packets they send from the secure network through the tunnel; they decrypt the packets they receive from the tunnel and route them to the destination hosts. In this scenario, the SAs do not authenticate the destination hosts (just the firewalls) and an attacker's modification of the IP headers could go undetected.

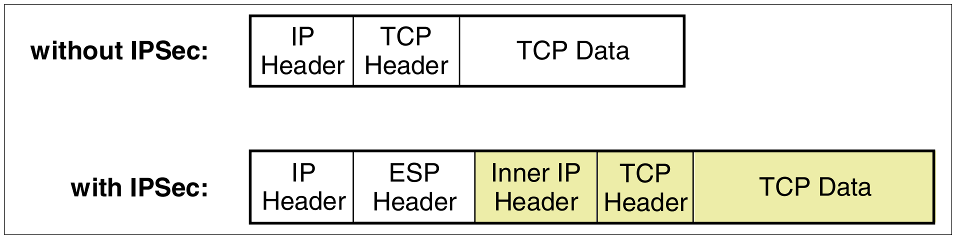


Figure 5-19 ESP in tunnel mode

In this environment, tunnel mode is to be used. Figure 5-19 shows the format of ESP packets in this mode; again, protected areas are shaded. The complete original packet, including the original IP header, is used as payload for an ESP packet. The inner IP header has the address of the destination host while the outer IP header addresses the firewall at the end of the tunnel. In this way, the complete packet including the IP header is protected.

In some cases, the AH and ESP formats are combined (applied one after the other) in order to reap both the benefits of IP header authentication with AH and payload (data) encryption with ESP.

For detailed information, read:

- *A Comprehensive Guide to Virtual Private Networks, Volume I: IBM Firewall, Server and Client Solutions,* SG24-5201
- *Secure e-business in TCP/IP Networks on OS/390 and z/OS*, SG24-5383

5.4.2 Alternative VPN solutions: Layer 2 Tunnelling Protocol

A remote access dial-up solution for mobile users is a very simple form of a virtual private network, typically used to support dial-in access to a corporate network whose users are all company employees. To eliminate the long-distance charges that would occur if a remote user were to dial in directly to a gateway on the home network, the IETF developed a tunneling protocol, Layer 2 Tunnel Protocol (L2TP). This protocol extends the span of a PPP connection: instead of beginning at the remote host and ending at a local ISP's point of presence, the virtual PPP link now extends from the remote host all the way back to the corporate gateway. In effect, the remote host appears to be on the same subnet as the corporate gateway.

Since the host and the gateway share the same PPP connection, they can take advantage of PPP's ability to transport protocols other than just IP. For example, L2TP tunnels can be used to support remote LAN access as well as remote IP access. Figure 5-20 outlines a basic L2TP configuration:

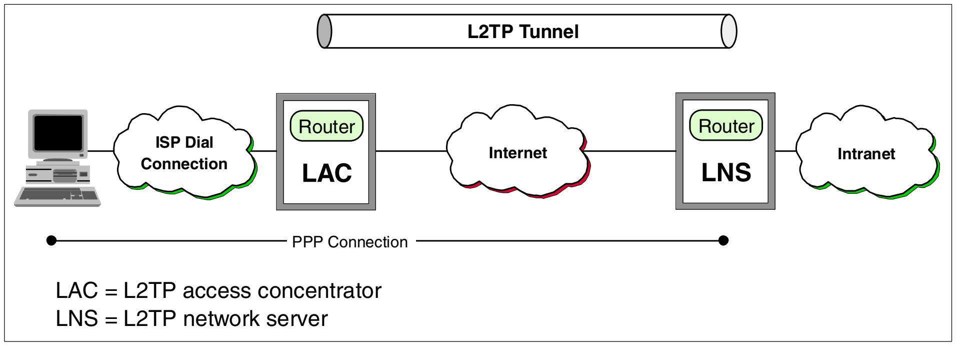


Figure 5-20 Layer 2 Tunnel Protocol (L2TP) scenario

Although L2TP provides cost-effective access, multi-protocol transport, and remote LAN access, it does not provide cryptographically robust security features. For example:

- Authentication is provided only for the identity of tunnel endpoints, but not for each individual packet that flows inside the tunnel. This can expose the tunnel to various attacks.
- Without per-packet integrity, it is possible to mount denial-of-service attacks by generating bogus control messages that can terminate either the L2TP tunnel or the underlying PPP connection.
- L2TP itself provides no facility to encrypt user data traffic. This can lead to embarrassing exposures when data confidentiality is an issue.
- While the payload of the PPP packets can be encrypted, the PPP protocol suite does not provide mechanisms for automatic key generation or for automatic key refresh. This can lead to someone listening in on the wire to finally break that key and gain access to the data being transmitted.

5.5 Secure Sockets Layer (SSL)

Secure Sockets Layer (SSL) is a protocol developed by the Netscape Communications Corporation that uses encryption to provide privacy and authentication between two applications using TCP/IP. SSL can be regarded as a *transport layer* equivalent of IPSec. Like IPSec, it uses asymmetric cipher algorithms (RSA is normally used) to authenticate users and sign messages, and symmetric algorithms to ensure confidentiality. Unlike IPSec, it is used to protect sessions between particular applications on particular ports; IPSec provides blanket protection between two hosts.

HTTP can use SSL to secure its communications. This allows Web browsers and servers to pass confidential or sensitive data through the Internet or intranet. SSL is also implemented by the Lightweight Directory Access Protocol (LDAP) for secure connections between LDAP clients and LDAP servers, by Telnet, and by a Telnet client such as Host On-Demand for connections between the client and the host system.

The use of SSL for Web access is through a protocol called HTTPS. HTTPS is a unique protocol that combines SSL and HTTP. You need to specify `https://` instead of `http://` as an anchor in HTML documents that link to SSL-protected documents. A client user can also open a URL by specifying `https://` to request SSL-protected documents.

Because HTTPS and HTTP are different protocols and use different ports (the default ports are 443 and 80, respectively), you can run both SSL and non-SSL requests at the same time. As a result, you can elect to provide information to all users using no security, and specific information only to browsers that make secure requests. This is how a retail company on the Internet can allow users to look through the merchandise without security, but then fill out order forms and send their credit card numbers using security.

SSL relies on digital certificates and a hierarchy of trusted authorities, as described in 5.2.5, “Digital signatures” on page 99, to ensure authentication of clients or servers.

5.5.1 Establishing secure communications with SSL

To use SSL, both the client and the server need to have the software to support this protocol. Because SSL started with HTTP communication, it is used as an illustration.

The latest Netscape and Microsoft browsers support SSL 3.0 and all its features on the client. SSL is composed of two sub-protocols:

- SSL Handshake Protocol
- SSL Record Protocol

The SSL Handshake Protocol initializes a secure session, with authentication of the server (and optionally, the client), agreement of encryption scheme, and transfer of encryption keys. A public-key algorithm, usually RSA, is used for the exchange of the symmetric encryption key and for digital signatures. With the server certificate, the client is also able to verify the server's identity. With SSL Version 3.0, the possibility of authenticating the client identity by using client certificates in addition to server certificates was added. The overall flow of these steps is shown in Figure 5-21.

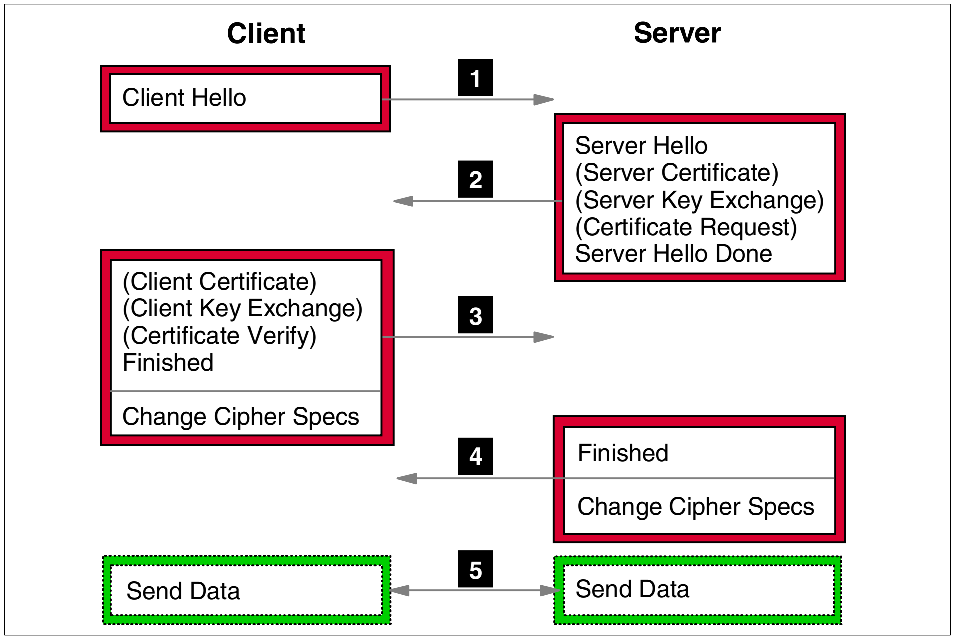


Figure 5-21 Overview of SSL Handshake Protocol

The SSL handshake process detailed in Figure 5-21 is explained in more detail in the following section:

- Step **1**: The client sends a connection request with a `client hello` message. This message includes:
  - Desired version number.
  - Time information (the current time and date in standard UNIX 32-bit format).
  - Optionally session-ID. If it is not specified the server will try to resume previous sessions or return an error message
  - Cipher suites. (List of the cryptographic options supported by the client. These are authentication modes, key exchange methods, encryptions and MAC algorithms.)
  - Compression methods supported by the client.
  - A random value (nonce). A nonce is a random value used in communication protocols, typically for replay protection.

- Step **2**: The server evaluates the parameters sent by the `client hello` message and returns a `server hello` message that includes the following parameters which were selected by the server to be used for the SSL session:
  - Version number
  - Time information (the current time and date in standard UNIX 32-bit format)
  - Session ID
  - Cipher suite
  - Compression method
  - A random value

  Following the `server hello` message, the server sends the following messages:
  - Server certificate if the server is required to be authenticated
  - A server key exchange message if there is no certificate available or the certificate is for signing only
  - A certificate request if the client is required to be authenticated

  Finally, the server sends a `server hello done` message and begins to wait for the client response.
- Step **3**: The client sends the following messages:
  - If the server has sent a certificate request, the client must send a certificate or a `no certificate` message.
  - If the server has sent a server key exchange message, the client sends a client key exchange message based on the public key algorithm determined with the hello messages.
  - If the client has sent a certificate, the client verifies the server certificate and sends a `certificate verify` message indicating the result.

  The client then sends a `finished` message indicating the negotiation part is completed. The client also sends a `change cipher spec` message to generate shared secrets. It should be noted that this is not controlled by the handshake protocol, the change cipher spec protocol manages this part of the operation.
- Step **4**: The server sends a `finished` message indicating the negotiation part is completed. The server then sends the `change cipher spec` message.
- Step **5**: Finally, the session partners separately generate an encryption key, in which they derive the keys to use in the encrypted session that follows from the master key. The Handshake protocol changes the state to the connection state. All data taken from the application layer is transmitted as special messages to the other party.

The SSL Record Protocol transfers application data using the encryption algorithm and keys agreed upon during the handshake phase. As explained above, symmetric encryption algorithms are used, because they provide much better performance than asymmetric algorithms.

5.5.2 SSL considerations

As discussed, security functions such as SSL are needed to send sensitive data safely if you connect your system to an insecure network such as the Internet. On the other hand, using such security functions has performance impacts, including utilizing additional CPU cycles and degrading Web server performance. There is significant additional overhead in starting up an SSL session compared with a normal HTTP connection. The protocol avoids some of this overhead by allowing the client and server to retain session key information and to resume that session without negotiating and authenticating a second time.

Furthermore, SSL does not satisfy every security requirement. While it protects against eavesdropping and alteration of data, it cannot protect the server from an attacker masquerading as a trusted user. For these security concerns, the risk can be minimized by the use of access controls or firewalls.

To maintain SSL security, you have to manage the key carefully, especially when using self-signed certificates, because the whole system environment is affected by the security of the Certificate Authority's key database.

5.5.3 Where to use SSL

Typically, SSL is used for communications between a client machine and hosts within a DMZ for the following reasons:

- Client authentication and authorization is to be performed within the DMZ using the client's password and digital certificate
- The data that is being transmitted must be encrypted (passwords, credit card numbers, personal data, and so on)

In a similar manner, SSL can be used for communications between hosts in the DMZ and hosts within the internal network for the following reasons:

- The data that is being transmitted must be encrypted for security purposes
- Client authentication and authorization is to be performed within the enterprise internal network using the client's password and digital certificate

Terminating and SSL session within the DMZ and performing client authentication and authorization there has the advantage of removing the burden of SSL processing and authorization/authentication processing from back-end application servers.

5.6 Authentication and authorization

Authentication is normally a prerequisite for authorization (unless everybody is authorized to do something, such as autonomous FTP), but they are separate and distinct concepts.

- Authentication establishes who you are.
- Authorization establishes what you are allowed to do.

5.6.1 User authentication

Authentication is the process of reliably verifying the identity of someone (or something). This is distinct from the assertion of identity (known as identification) and from deciding what privileges accrue to that identity (authorization). Authentication is the most difficult from the perspective of network security. Classically, there are three different ways that you can authenticate yourself or a computer to another computer system:

1. You can tell the computer something that you know; for example, a password. This is the traditional password system.
2. You can “show” the computer something you have; for example, a digital certificate, a card key, a smart card, one-time pads, a challenge-response list, and so on.
3. You can let the computer measure something about you; for example, your fingerprint, a retina scan, voiceprint analysis and so on.

Some systems combine these approaches. For example, a smart card that requires the user to enter a personal identification number (PIN) to unlock it is a combination of something you have (the card) and something you know (the PIN). In theory, it is considered a good idea to combine at least two mechanisms, because people can steal either one: the thing you have is susceptible to ordinary theft, and the thing you know is compromised by sniffing if it passes over the Internet; but it is rare for somebody to be able to get both at once. Automatic teller machines use this combination; however, ATMs also demonstrate the flaw in the theory: when you are authenticating (standing at the ATM), you reveal what you have (your card) and what you know (your PIN) simultaneously, making yourself vulnerable to a thief who watches you use the machine to capture your PIN, then steals your card as you leave.

In an e-business application server environment, a common approach to user and server authentication is to implement SSL and digital certificates to “show the computer something you have” while also using passwords to tell the computer “something you know”.

5.6.2 User authorization

Authorization is verifying that the person (or machine) is really allowed to do what it is requesting to do. For example, does the user have the right to invoke a method on an EJB or access a particular HTML page, servlet or JavaServer Pages (JSP)? Authorization is usually checked after user authentication. Authorization is achieved by assigning access controls, such as read, write, or delete, for user IDs or group names to the objects (EJBs, HTML pages, servlets, and so on) being accessed. These controls, along with the authorized users or groups, can be maintained in an access control list (ACL) associated with each object.

For example, assume a Web server has authenticated user COMPU23. Now it needs to decide whether to let COMPU23 do something, for instance read a particular HTML document stored on that computer. The ACL for that HTML file specifies who is allowed to use the file and in what way (for example, read, write, execute, delete). Once you prove (via authentication) that you are COMPU23, you are allowed access to any resource that lists COMPU23 in its ACL.

Maintaining ACLs on every object to be controlled can quickly become prohibitively expensive. For example, imagine an object that all IBM employees are allowed to see. The ACL associated with that file would be very large. Furthermore, there might be many objects that should be accessible to all IBM employees and a few objects that should be accessible to only a few. The person maintaining the ACLs for all those objects would be very busy adding and deleting entries for each of the ACLs and remembering which objects specific employees have access to.

A common solution for this level of authorization is to introduce the concept of groups. Instead of specifying all the individuals on every file, have a separate file called something like IBMEmployees, and have the ACL for each object (intended for that group) have an entry IBMEmployees.

However, there still could be some inefficiency. There might be objects (files, servlets, HTML pages, and so on) on several different Web servers, all intended to be accessed by all the employees at IBM. If the system managers of the various systems talked to one another, they would realize they were duplicating effort, since every one of them had to maintain a file listing all individuals in the IBM corporation. There are two possible methods of eliminating this duplication of effort.

- The first one involves storing the file IBMEmployees at some central location, and having every system query the central location when deciding whether to grant a given user access to a resource whose ACL lists IBMEmployees.
- The second method is to include information about what groups an individual belongs to inside the individual's digital certificate. So COMPU23's certificate would specify that he was a member of IBMEmployees, and any other groups that he was a member of (such as Engineers or Programmers).

What are the trade-offs between these two approaches? In the first approach, evaluation of an ACL (that is, finding out if a particular individual should be granted access to a resource that lists IBMEmployees in its ACL) requires sending a message to the central location.

The second approach has issues as well. Security policies may mandate the invention of hundreds or thousands of groups (employees who have been hired less than a year ago, employees who have special disabilities, employees who are working on a specific project, and so on). In that case, an individual's digital certificate can get very large. In addition, maintaining and distributing a digital certificate for each individual that accurately represents that person's group status becomes much more cumbersome.

The approach we take with the Runtime pattern that follows is to implement a central location that maintains security policy user access control lists.

5.7 Authenticating proxy

The Tivoli Access Manager plug-in for WebSphere Edge Server V2 is an authentication and authorization plug-in for the Edge Server Caching Proxy component. The plug-in provides an "edge of network" authentication and authorization point for all users wishing to access resources within a domain secured by Tivoli Access Manager .

The Tivoli Access Manager plug-in performs user authentication based on HTTP Authenticate headers in a reverse proxy configuration. In a Tivoli Access Manager domain, where the Tivoli Access Manager plug-in is installed as part of a reverse proxy, no other content or application server in the domain can perform its own user authentication. Users authenticated through the Tivoli Access Manager plug-in can only access content inside the Tivoli Access Manager domain.

5.7.1 Runtime pattern variation 8: Authenticating proxy

In this Runtime pattern a proxy server layer is implemented in front of all content server nodes to provide security services at the edge of your network. A plug-in, the Tivoli Access Manager plug-in, is used to authenticate and authorize requests to the site. This reduces the amount of security processing load from the application server nodes, freeing resources to do application processing work.

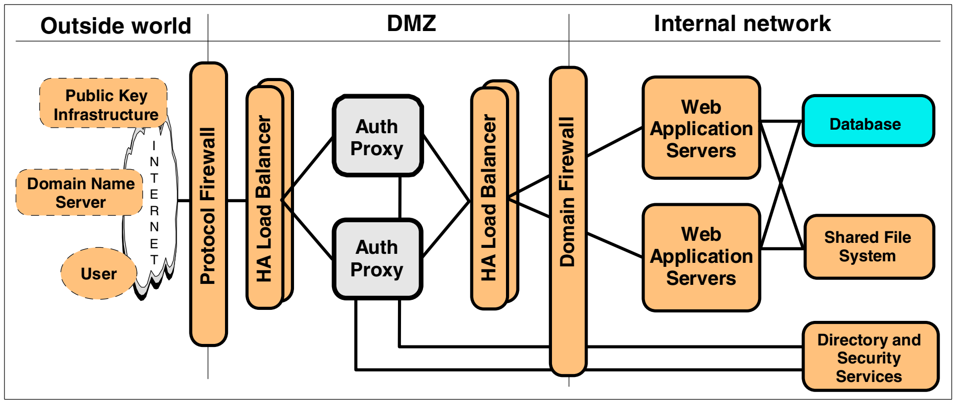


Figure 5-22 Runtime pattern 8- Authenticating Proxy

Benefits:

Runtime pattern 8 offers the following benefits:

- ► It reduces the load on application server nodes.
- ► Scalability is improved by shifting load to the edge of the network.
- ► Centralized security management can be achieved.

Disadvantages:

This Runtime pattern has some disadvantages:

- ► Additional resources are needed.
- ► Complex planning and installation requires more lead time.
- ► A broad range of in-depth security product knowledge is required.

Part 3

Runtime implementation

6

Common installation and configuration guidelines

This chapter provides a short guide to the installation of the WebSphere Edge Server features used in the Runtime patterns of this book. We recommend that you familiarize yourself with the product documentation, as we do not cover all aspects of customizing all features to your environment and policies.

6.1 Common installation guidelines

More detailed product installation information can be found in the following publications:

- *WebSphere Edge Server for Multiplatforms Getting Started Version 2*, SC09-4566
- *WebSphere Edge Server for Multiplatforms Network Dispatcher Administration Guide Version 2*, GC31-8496
- *IBM WebSphere Edge Server: New Features and Functions in Version 2*, SG24-6511.

6.1.1 General guidelines

1. Verify that you have enough disk space available.
2. Log on as root/Administrator.
3. If you are accessing a Unix server remotely, export the display variable with your local machines IP address:

   ```
   export DISPLAY=192.168.10.104:0
   ```

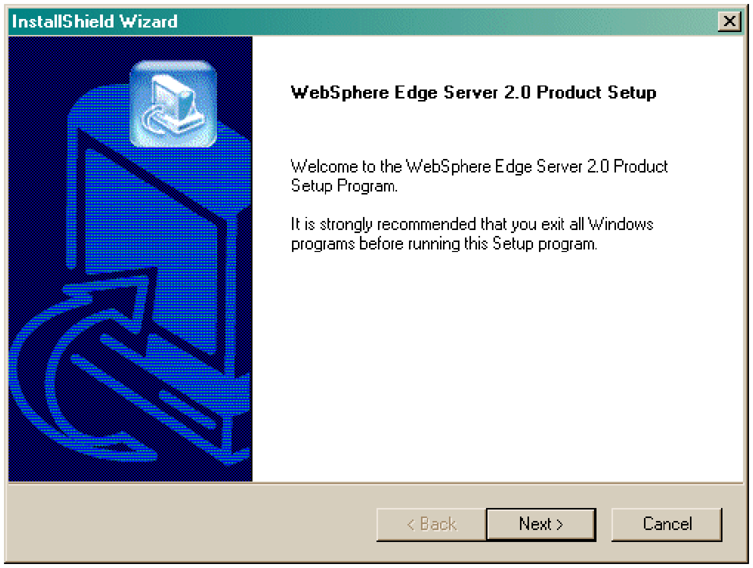
4. Insert the WebSphere Edge Server Product CD into the server's CD ROM drive.
5. On AIX and Linux, mount the CD. For AIX the command is:

   ```
   mount -rv cdrfs /dev/cd0 /mnt
   ```

6. Change directory to the CD and start the setup/install program.

6.1.2 Load Balancer

1. Start the setup.exe or install.sh program. The welcome window shown in Figure 6-1 will be displayed.



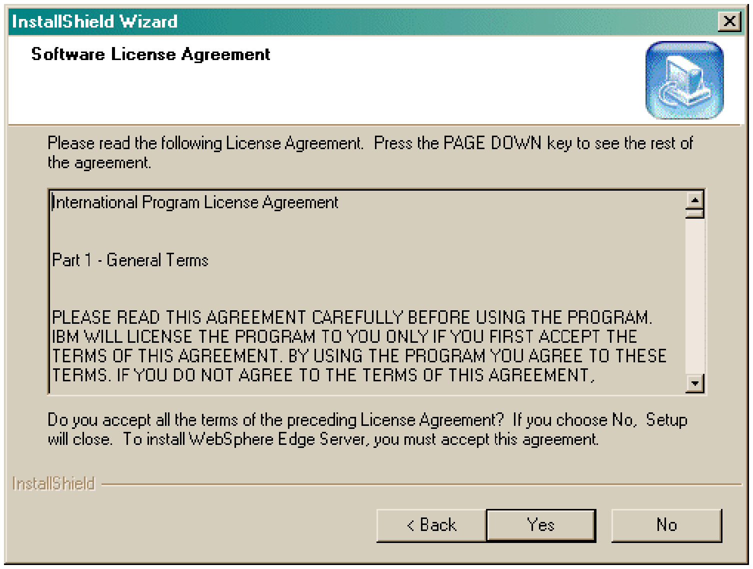
WebSphere Edge Server 2.0 Product Setup

Welcome to the WebSphere Edge Server 2.0 Product Setup Program.

It is strongly recommended that you exit all Windows programs before running this Setup program.

Figure 6-1 Welcome window

2. Click **Next** to start the installation.



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

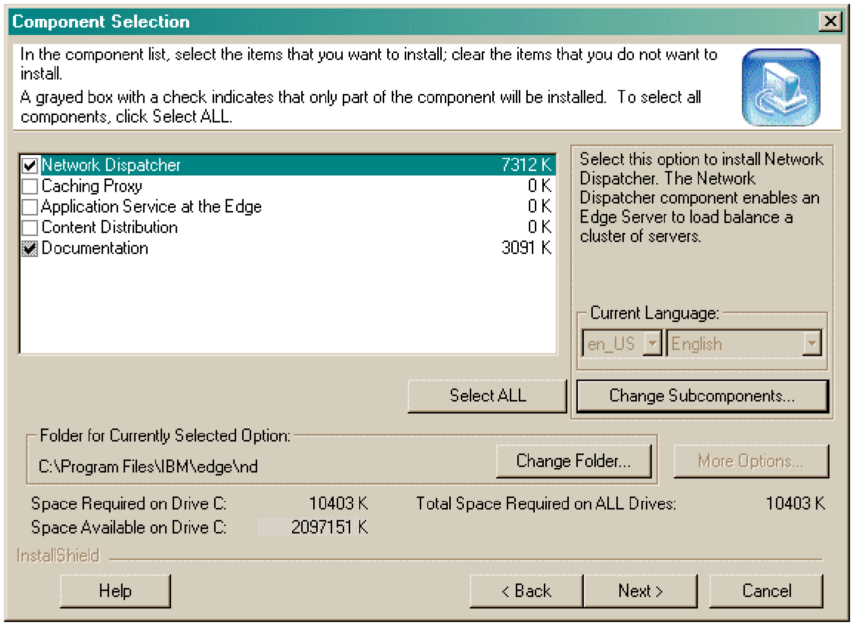
Part 1 - General Terms

PLEASE READ THIS AGREEMENT CAREFULLY BEFORE USING THE PROGRAM. IBM WILL LICENSE THE PROGRAM TO YOU ONLY IF YOU FIRST ACCEPT THE TERMS OF THIS AGREEMENT. BY USING THE PROGRAM YOU AGREE TO THESE TERMS. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT,

Do you accept all the terms of the preceding License Agreement? If you choose No, Setup will close. To install WebSphere Edge Server, you must accept this agreement.

Figure 6-2 Software License Agreement

3. Accept the Software Licence Agreement by clicking **Yes** in the window shown in Figure 6-2.


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.

✓ Network Dispatcher	7312 K
Caching Proxy	0 K
Application Service at the Edge	0 K
Content Distribution	0 K
✓ Documentation	3091 K

Select this option to install Network Dispatcher. The Network Dispatcher component enables an Edge Server to load balance a cluster of servers.

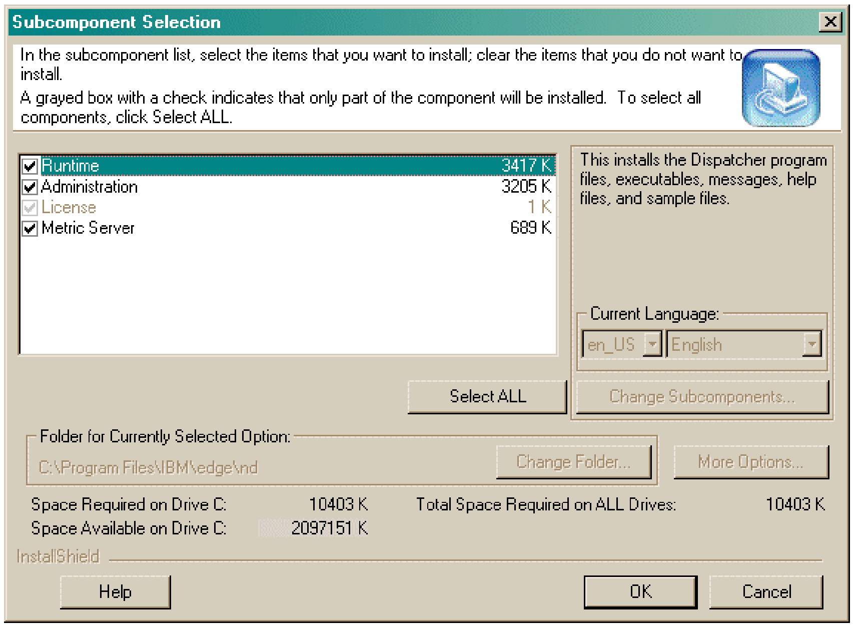
Current Language: en_US English

Folder for Currently Selected Option:
C:\Program Files\IBM\edge\nd

Space Required on Drive C: 10403 K
Space Available on Drive C: 2097151 K
Total Space Required on ALL Drives: 10403 K

Figure 6-3 Component selection window

4. Select the **Network Dispatcher** component in the Component Selection window. The default selection will install the Load Balancer Runtime and Administrative component, Metric Server and Documentation. To change the selection, click **Change Subcomponents** (see Figure 6-3). When you return to this window, click **Next** to proceed.


Subcomponent Selection

In the subcomponent 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.

☑ Runtime	3417 K
☑ Administration	3205 K
☑ License	1 K
☑ Metric Server	689 K

This installs the Dispatcher program files, executables, messages, help files, and sample files.

Current Language: en_US English

Folder for Currently Selected Option: C:\Program Files\IBM\edge\nd

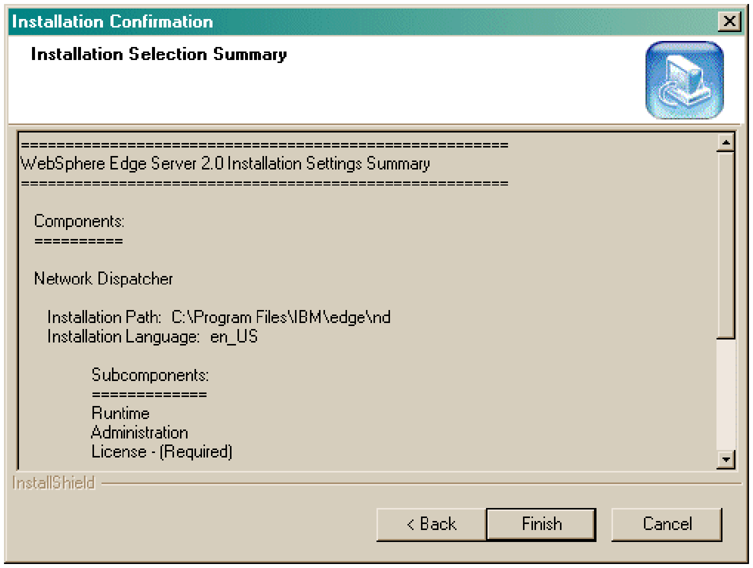
Space Required on Drive C: 10403 K

Space Available on Drive C: 2097151 K

Total Space Required on ALL Drives: 10403 K

Figure 6-4 Subcomponent selection window

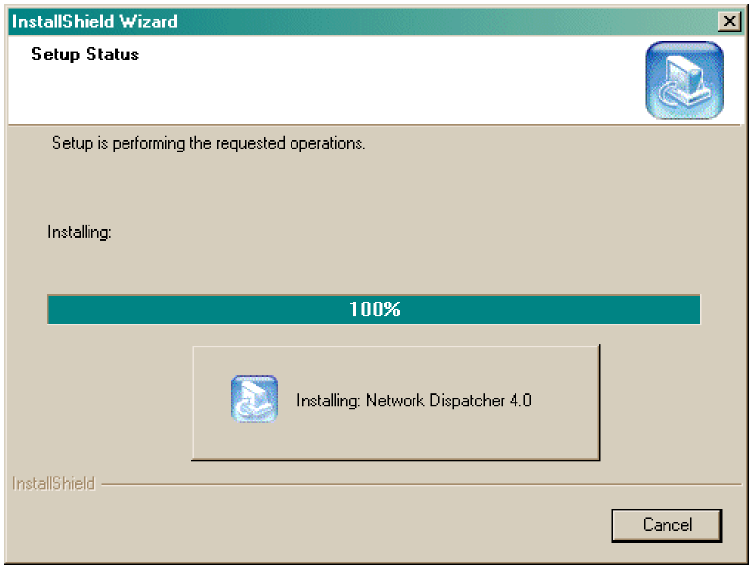
5. At the very least, you will require the Runtime component and License. Typically, the Metric Server component is installed on back-end server machines in a Site Selector environment, or load balanced Load Balancers in a Wide Area Network Dispatcher or Site Selector environment. After you have made your selections, click **OK**.



```
=======================================================
WebSphere Edge Server 2.0 Installation Settings Summary
=======================================================

  Components:
  ==========

  Network Dispatcher

    Installation Path:  C:\Program Files\IBM\edge\nd
    Installation Language:  en_US

           Subcomponents:
           =============
           Runtime
           Administration
           License - (Required)
```

Figure 6-5 Selection summary window

6. The window shown in Figure 6-5 displays a summary of the selected components to install. If no changes are required, click **Finish** to start the installation process.



Setup is performing the requested operations.

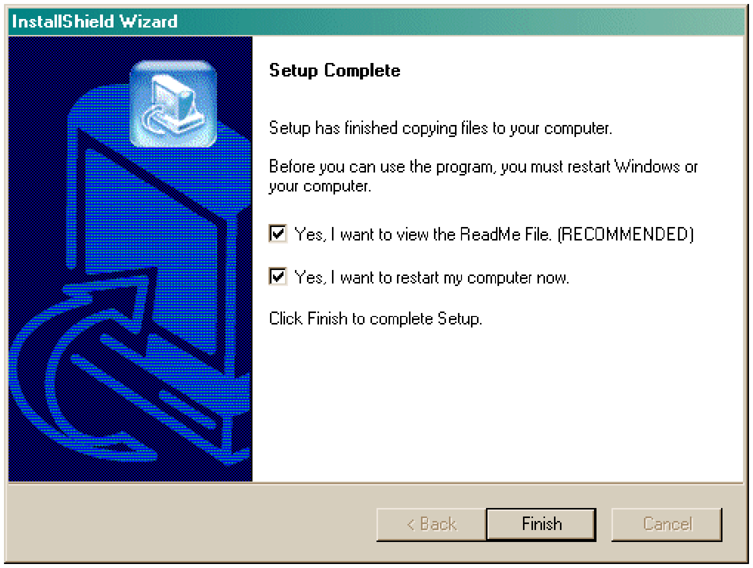
Installing:

100%

Installing: Network Dispatcher 4.0

Figure 6-6 Installation status window

7. The status window shown in Figure 6-6 highlights the progress of the installation. You may cancel the installation by clicking the **Cancel** button at any time.


InstallShield Wizard

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.

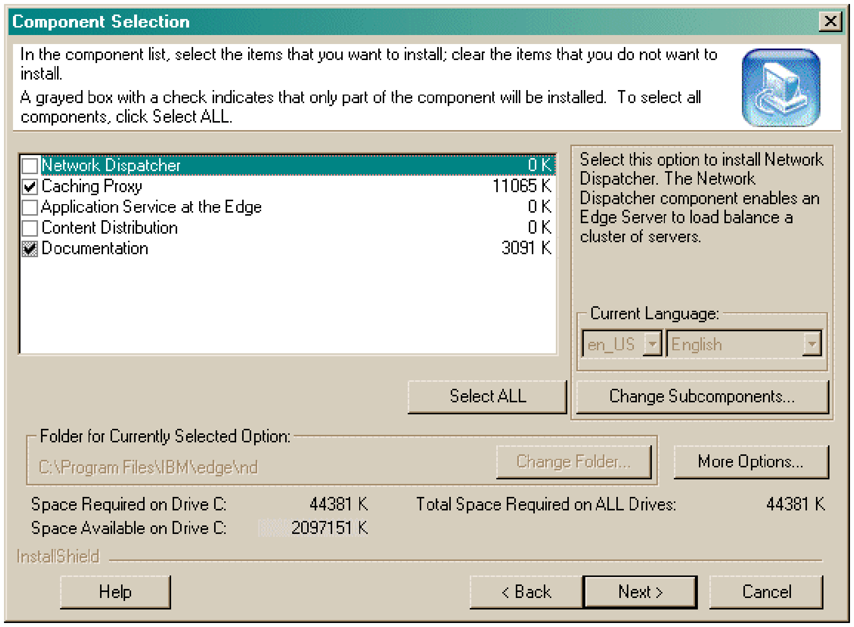
Click Finish to complete Setup.

Figure 6-7 Installation completion window

8. Once the install process has finished, you may view the Readme file and (on Windows) immediately reboot your server.

6.1.3 Caching Proxy

1. Start the setup.exe or install.sh program. The welcome window shown in Figure 6-1 will be displayed.
2. Click **Next** to start the installation.
3. Accept the Software Licence Agreement by clicking **Yes** in the window shown in Figure 6-2.



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.

☐ Network Dispatcher	0 K
☑ Caching Proxy	11065 K
☐ Application Service at the Edge	0 K
☐ Content Distribution	0 K
☑ Documentation	3091 K

Select this option to install Network Dispatcher. The Network Dispatcher component enables an Edge Server to load balance a cluster of servers.

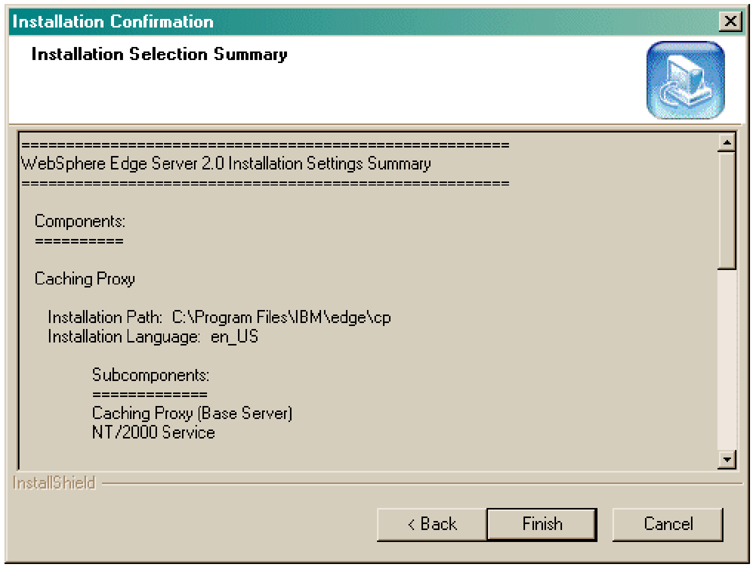
Current Language: en_US English

Folder for Currently Selected Option: C:\Program Files\IBM\edge\nd

Space Required on Drive C: 44381 K
Space Available on Drive C: 2097151 K
Total Space Required on ALL Drives: 44381 K

Figure 6-8 Component selection window

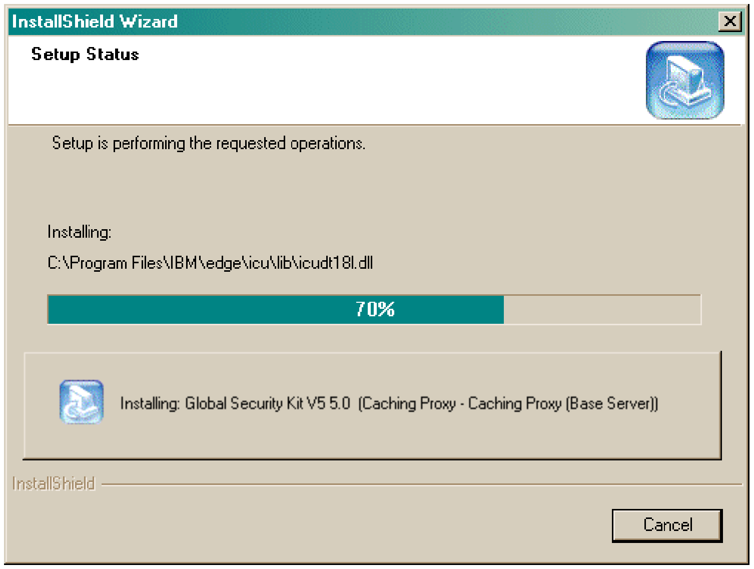
4. In the Component selection window, select the **Caching Proxy** checkbox. This will automatically select the Application Service at the Edge and Content Distribution checkboxes too. For a simple Caching Proxy, deselect those items. Click **Next** to proceed.



```
=======================================================
WebSphere Edge Server 2.0 Installation Settings Summary
=======================================================

  Components:
  ==========

  Caching Proxy

    Installation Path:  C:\Program Files\IBM\edge\cp
    Installation Language:  en_US

         Subcomponents:
         =============
         Caching Proxy (Base Server)
         NT/2000 Service
```

Figure 6-9   Component selection summary window

5. The window shown in Figure 6-9 displays a summary of the selected components to install. If no changes are required, click **Finish** to start the installation process.



Setup is performing the requested operations.

Installing:

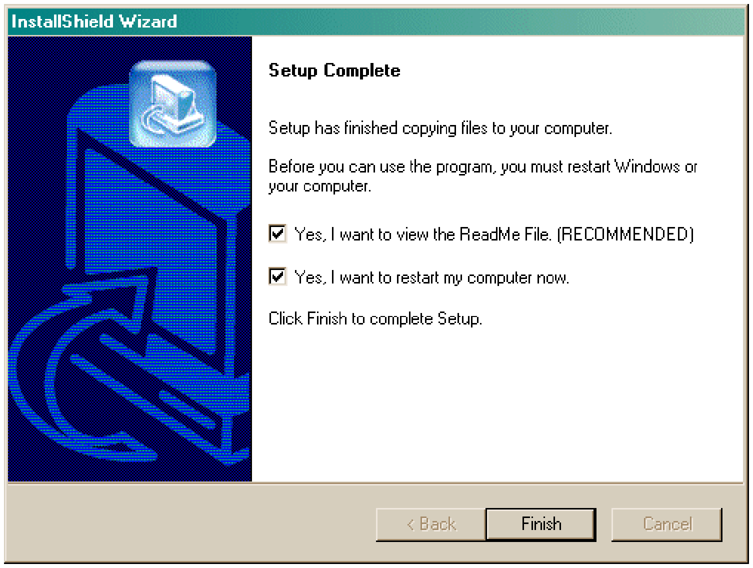
C:\Program Files\IBM\edge\icu\lib\icudt18l.dll

70%

Installing: Global Security Kit V5 5.0 (Caching Proxy - Caching Proxy (Base Server))

Figure 6-10   Installation status window

6. The status window shown in Figure 6-10 highlights the progress of the installation.



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 6-11 Installation completion window

7. Once the install process has finished, you may view the Readme file and (on Windows) immediately reboot your server.

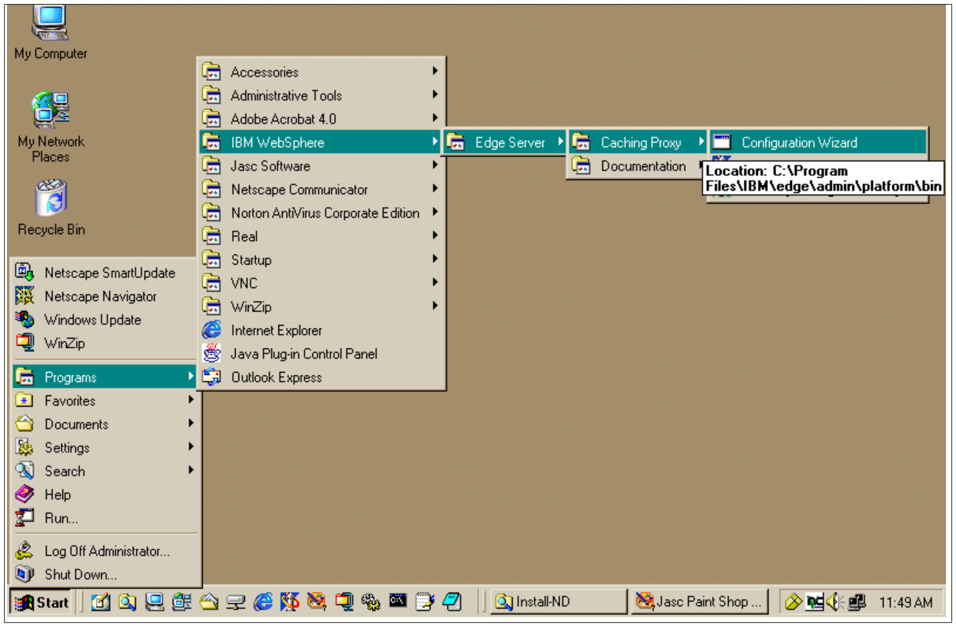
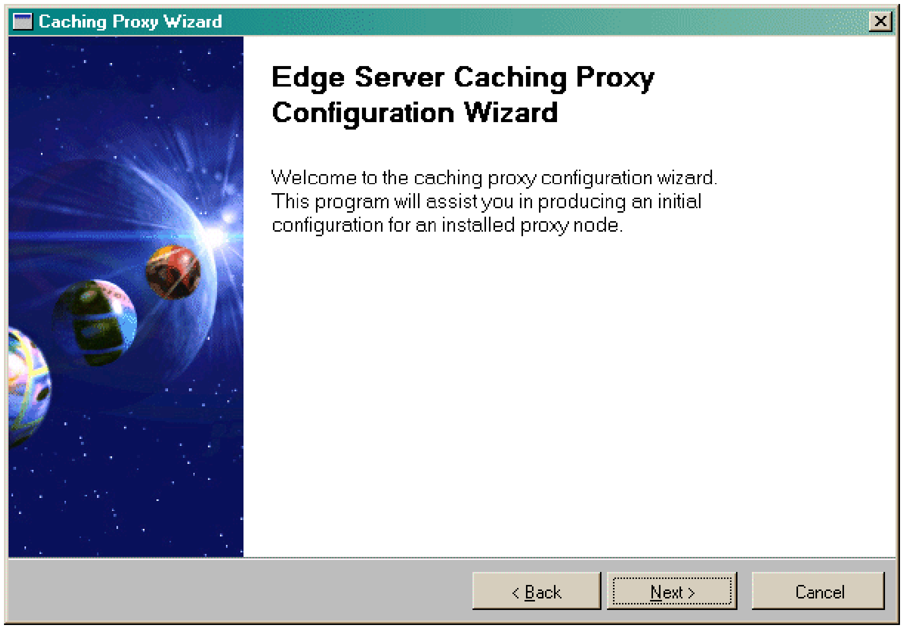


Figure 6-12 Start Caching Proxy configuration

8. To start the configuration of the Caching Proxy on Windows 2000, go to **Start -> Program -> IBM WebSphere -> Edge Server -> Caching Proxy -> Configuration Wizard** (see Figure 6-12). This will display the Caching Proxy Configuration Wizard window shown in Figure 6-13.

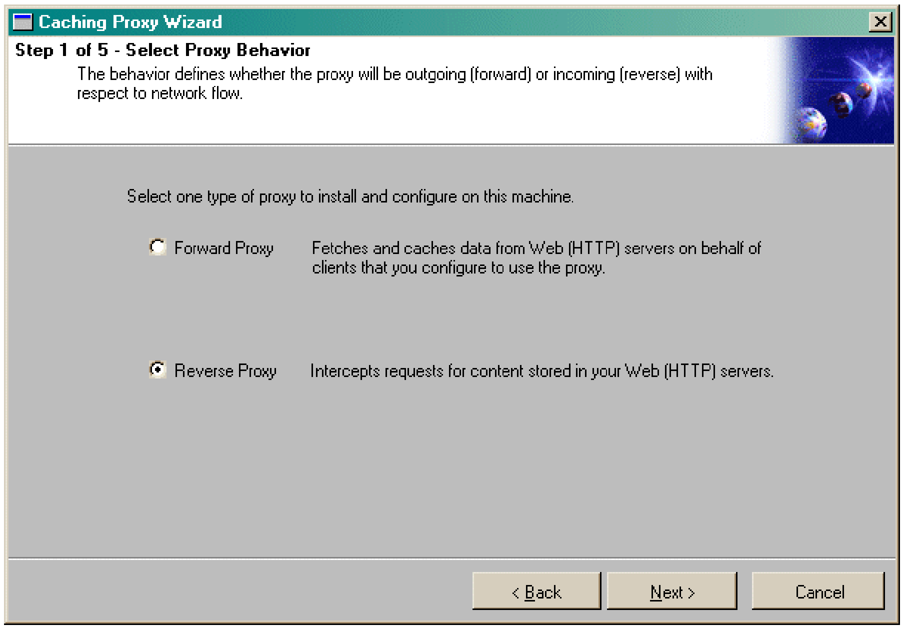

Caching Proxy Wizard

Edge Server Caching Proxy Configuration Wizard

Welcome to the caching proxy configuration wizard. This program will assist you in producing an initial configuration for an installed proxy node.

Figure 6-13 Caching Proxy Wizard window

9. To start the Configuration Wizard, click **Next**.



Step 1 of 5 - Select Proxy Behavior

The behavior defines whether the proxy will be outgoing (forward) or incoming (reverse) with respect to network flow.

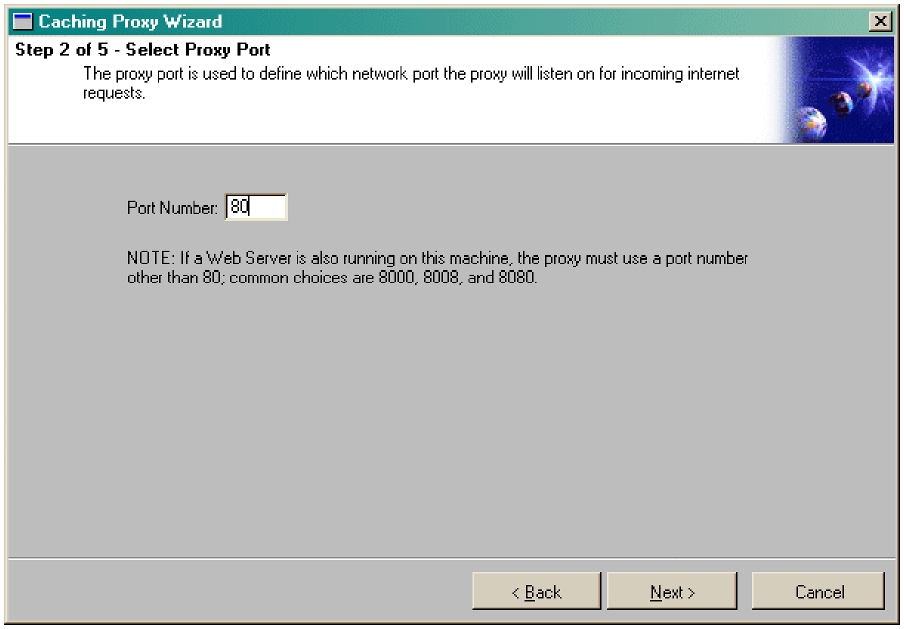
Select one type of proxy to install and configure on this machine.

Forward Proxy — Fetches and caches data from Web (HTTP) servers on behalf of clients that you configure to use the proxy.

[✓ Reverse Proxy] — Intercepts requests for content stored in your Web (HTTP) servers.

Figure 6-14 Step 1 - Select Proxy type

10. In this book, we always use the Caching Proxy as a reverse proxy. Select **Reverse Proxy** as shown in Figure 6-14 and click **Next**.


Caching Proxy Wizard

Step 2 of 5 - Select Proxy Port

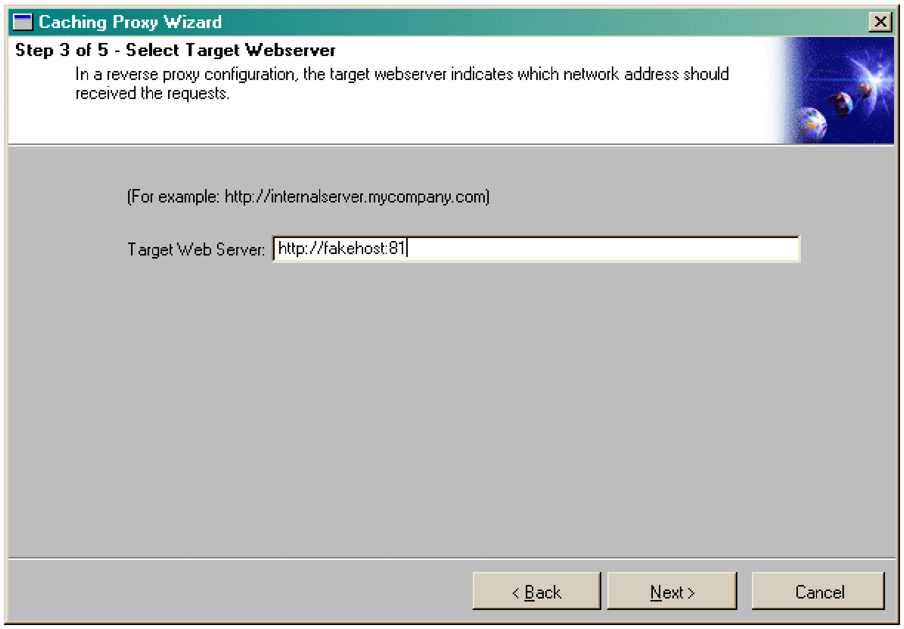
The proxy port is used to define which network port the proxy will listen on for incoming internet requests.

Port Number: 80

NOTE: If a Web Server is also running on this machine, the proxy must use a port number other than 80; common choices are 8000, 8008, and 8080.

Figure 6-15 Step 2 - Port selection

11. Select the port on which the proxy will be accessed. If there is an HTTP server running on the Caching Proxy machine that listens on port 80, you must select a different port for the Caching Proxy to listen on. Click **Next**.


Caching Proxy Wizard

Step 3 of 5 - Select Target Webserver

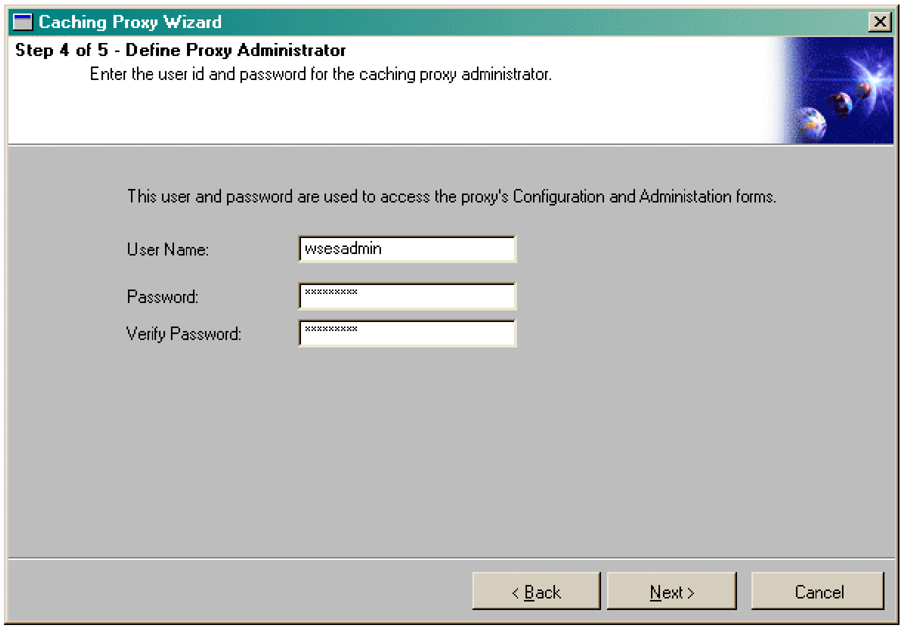
In a reverse proxy configuration, the target webserver indicates which network address should received the requests.

(For example: http://internalserver.mycompany.com)

Target Web Server: http://fakehost:81

Figure 6-16 Step 3 - Define the target server

12. Define the target back-end server that the Caching Proxy will access. Click **Next** to proceed.



Enter the user id and password for the caching proxy administrator.

This user and password are used to access the proxy's Configuration and Administation forms.

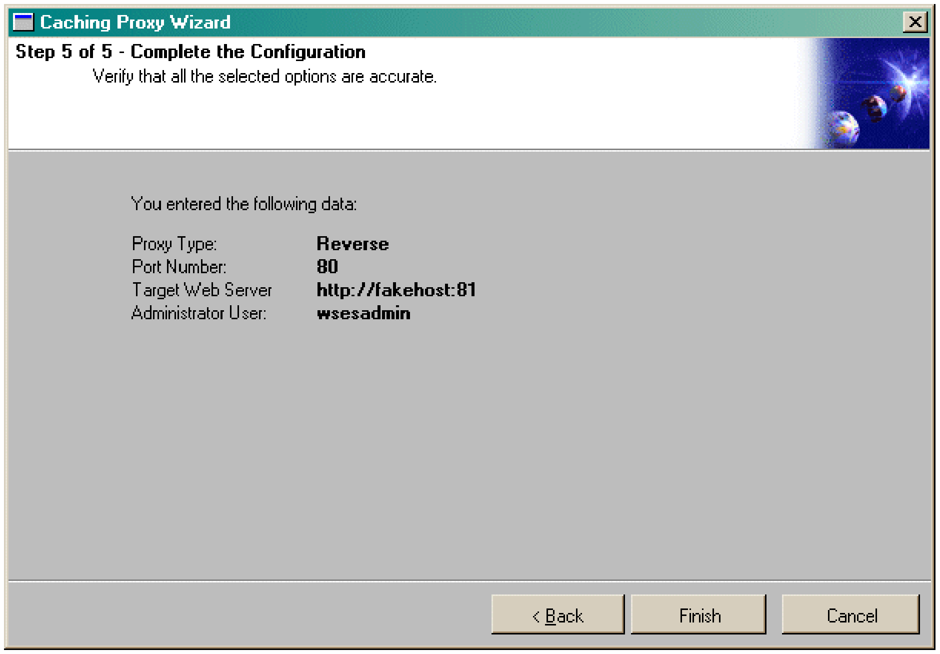
User Name: wsesadmin

Password: *********

Verify Password: *********

Figure 6-17 Step 4 - Define administrator account

13. Define a user name and password to administer the Caching Proxy through the Web-based Configuration and Administration forms. Click **Next**.


Caching Proxy Wizard
Step 5 of 5 - Complete the Configuration
Verify that all the selected options are accurate.
< Back
Finish
Cancel

You entered the following data:

Proxy Type:	Reverse
Port Number:	80
Target Web Server	http://fakehost:81
Administrator User:	wsesadmin

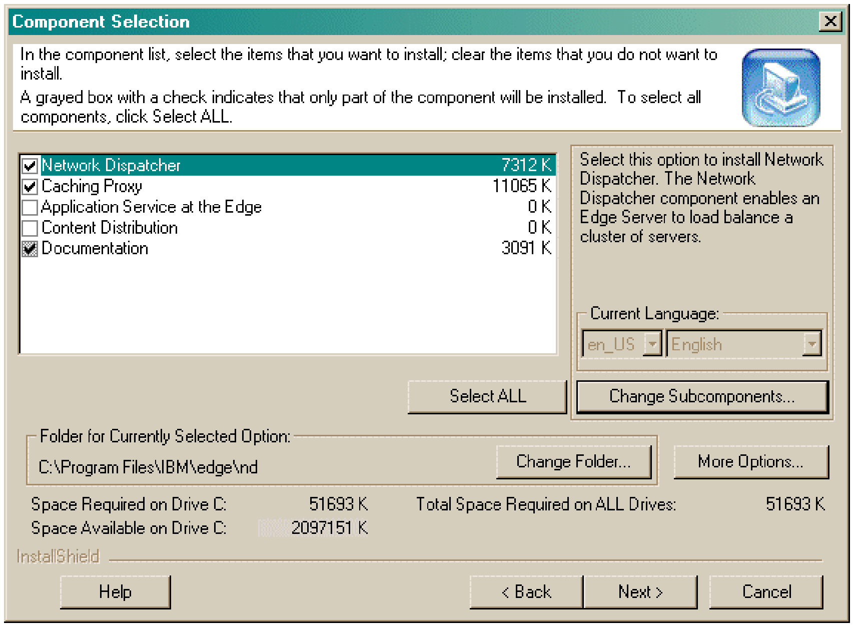
Figure 6-18 Step 5 - Summary

14. The window shown in Figure 6-18 presents a summary of your definitions. If they are as you wish, click **Finish**. Otherwise, you may go back and adjust the settings. The configuration will only take a moment to complete.

Note: In a reverse proxy configuration, the Configuration Wizard will enter the Proxy statement as the very first mapping rule. If you want to access the Caching Proxy Configuration and Administration forms from a browser to configure the Caching Proxy, you must move this statement by manually editing the ibmproxy.conf file on the Caching Proxy machine. Refer to Chapter 13, *Build a reverse caching proxy network*, in *WebSphere Edge Server for Multiplatforms Getting Started Version 2*, SC09-4566, or Chapter 4, *Configuring the basic proxy functions* in *WebSphere Edge Server for Multiplatforms Administration Guide Version 2*, GC09-4567 for more information.

6.1.4 Proxy CBR

1. Start the setup.exe or install.sh program. The welcome window shown in Figure 6-1 will be displayed.
2. Click **Next** to start the installation.
3. Accept the Software Licence Agreement by clicking **Yes** in the window shown in Figure 6-2.



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.

☑ Network Dispatcher	7312 K
☑ Caching Proxy	11065 K
☐ Application Service at the Edge	0 K
☐ Content Distribution	0 K
☑ Documentation	3091 K

Select this option to install Network Dispatcher. The Network Dispatcher component enables an Edge Server to load balance a cluster of servers.

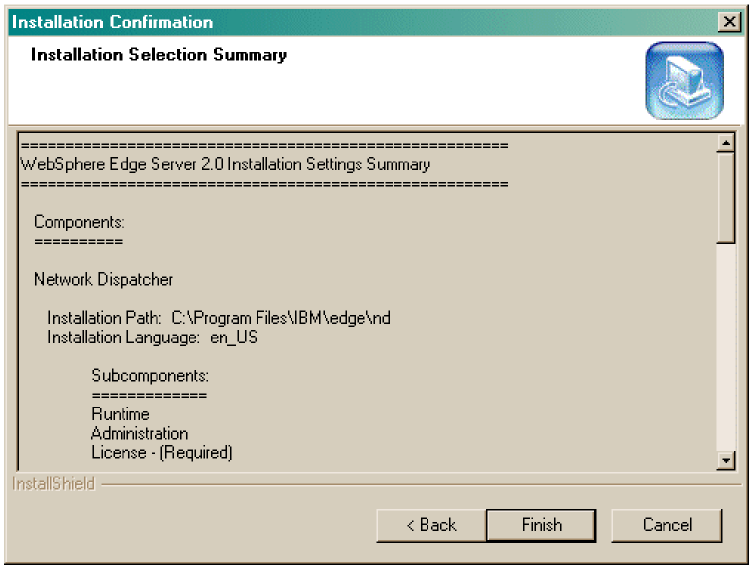
Current Language: en_US English

Folder for Currently Selected Option: C:\Program Files\IBM\edge\nd

Space Required on Drive C: 51693 K
Space Available on Drive C: 2097151 K
Total Space Required on ALL Drives: 51693 K

Figure 6-19 Component selection window

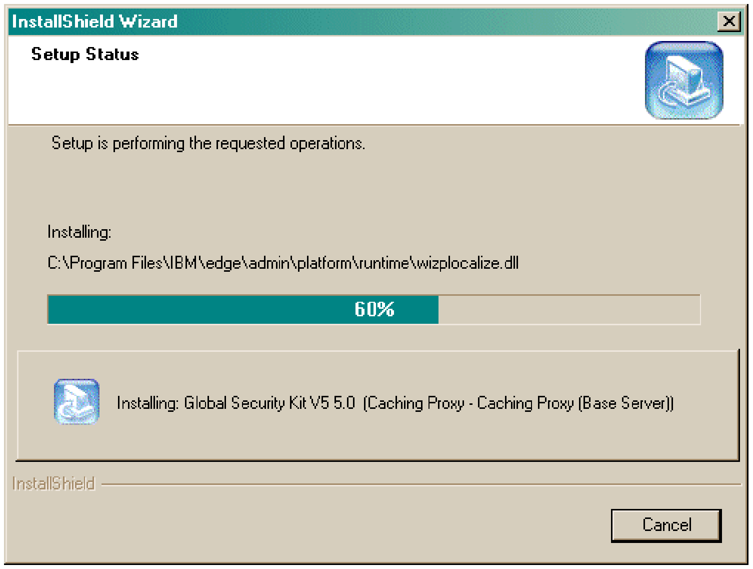
4. Select the **Load Balancer** and the **Caching Proxy** components. You will not need the Application Service at the Edge and the Content Distribution components. Click **Next** to start the installation.



```
=====================================================
WebSphere Edge Server 2.0 Installation Settings Summary
=====================================================

  Components:
  ==========

  Network Dispatcher

    Installation Path:  C:\Program Files\IBM\edge\nd
    Installation Language:  en_US

           Subcomponents:
           ============
           Runtime
           Administration
           License - (Required)
```

Figure 6-20 Component selection summary

5. This window shows you the selected components. Click **Finish** to start the installation.



Setup is performing the requested operations.

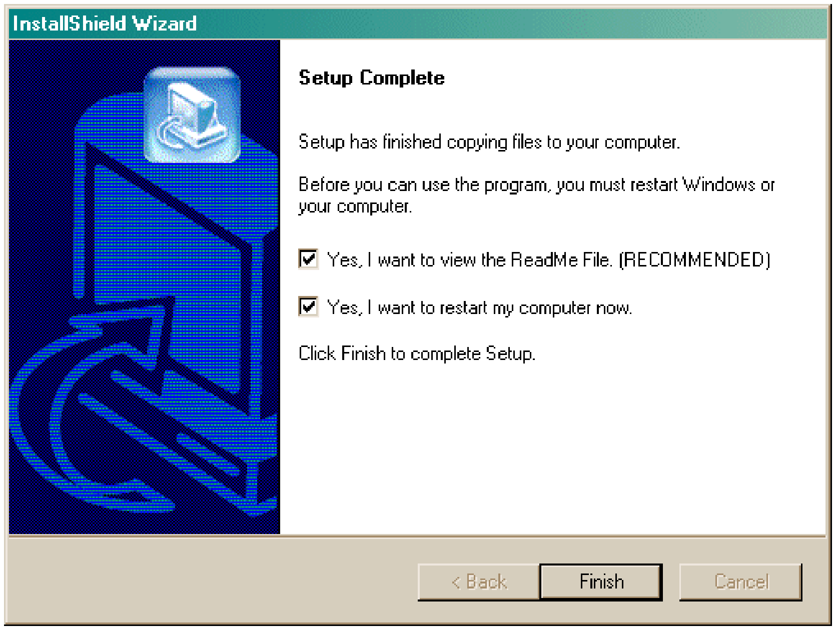
Installing:

C:\Program Files\IBM\edge\admin\platform\runtime\wizplocalize.dll

Installing: Global Security Kit V5 5.0 (Caching Proxy - Caching Proxy (Base Server))

Figure 6-21 Installation status window

6. The status window shows the progress of the installation.


InstallShield Wizard

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 6-22 Installation completion window

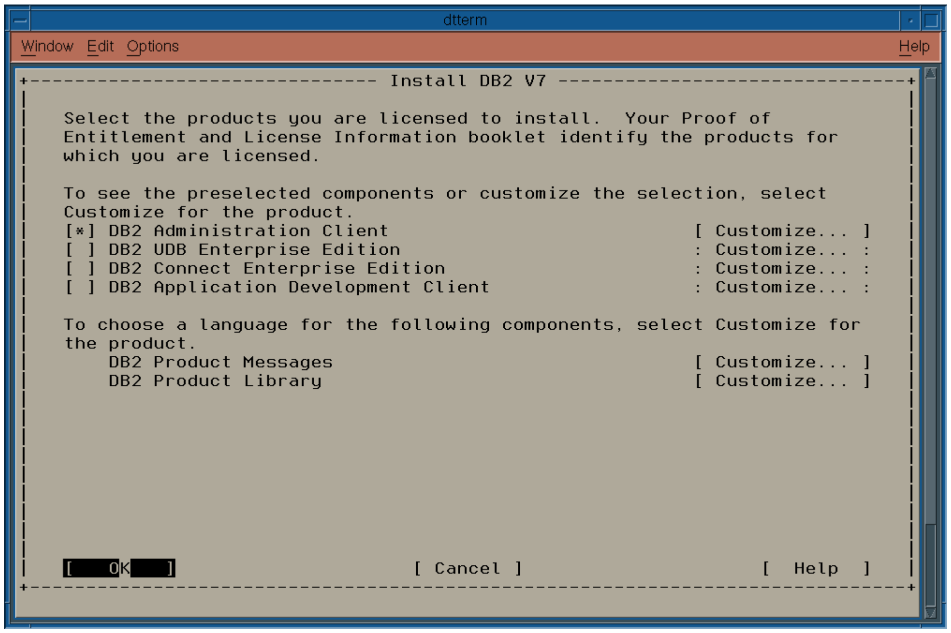
7. Once the install process has finished, you may view the Readme file and (on Windows) immediately reboot your server.

6.1.5 DB2 Client

1. Start the db2setup program:

   ```
   ./db2setup
   ```

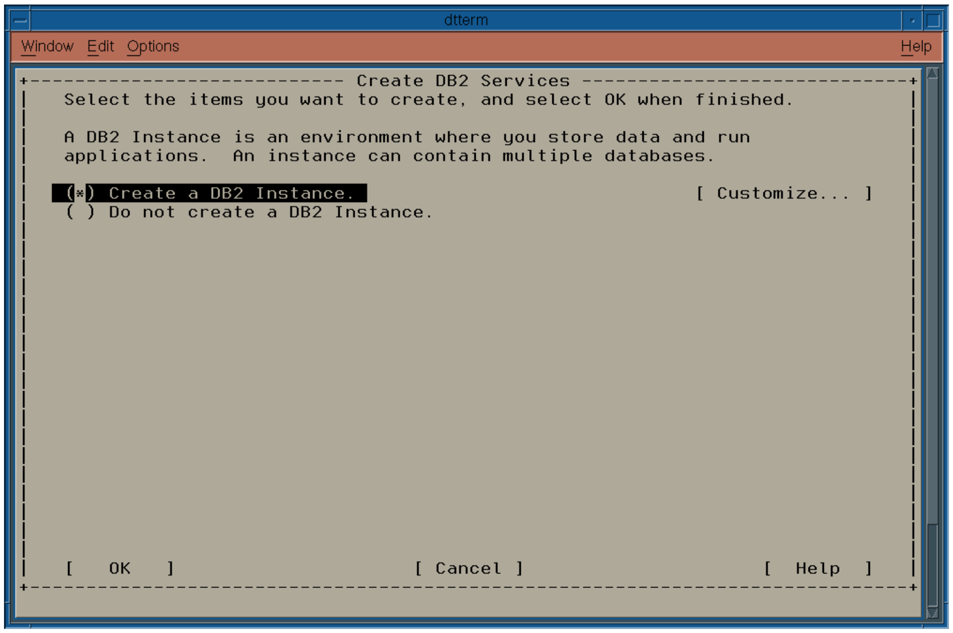
   This will result in the screen in Figure 6-23 being displayed.



```
+------------------------------- Install DB2 V7 --------------------------------+
|                                                                               |
|   Select the products you are licensed to install.  Your Proof of            |
|   Entitlement and License Information booklet identify the products for      |
|   which you are licensed.                                                    |
|                                                                               |
|   To see the preselected components or customize the selection, select      |
|   Customize for the product.                                                 |
|   [*] DB2 Administration Client                        [ Customize... ]      |
|   [ ] DB2 UDB Enterprise Edition                       : Customize... :      |
|   [ ] DB2 Connect Enterprise Edition                   : Customize... :      |
|   [ ] DB2 Application Development Client               : Customize... :      |
|                                                                               |
|   To choose a language for the following components, select Customize for   |
|   the product.                                                               |
|       DB2 Product Messages                             [ Customize... ]      |
|       DB2 Product Library                              [ Customize... ]      |
|                                                                               |
|   [    OK    ]              [ Cancel ]                    [  Help  ]        |
+-------------------------------------------------------------------------------+
```

Figure 6-23 db2setup screen

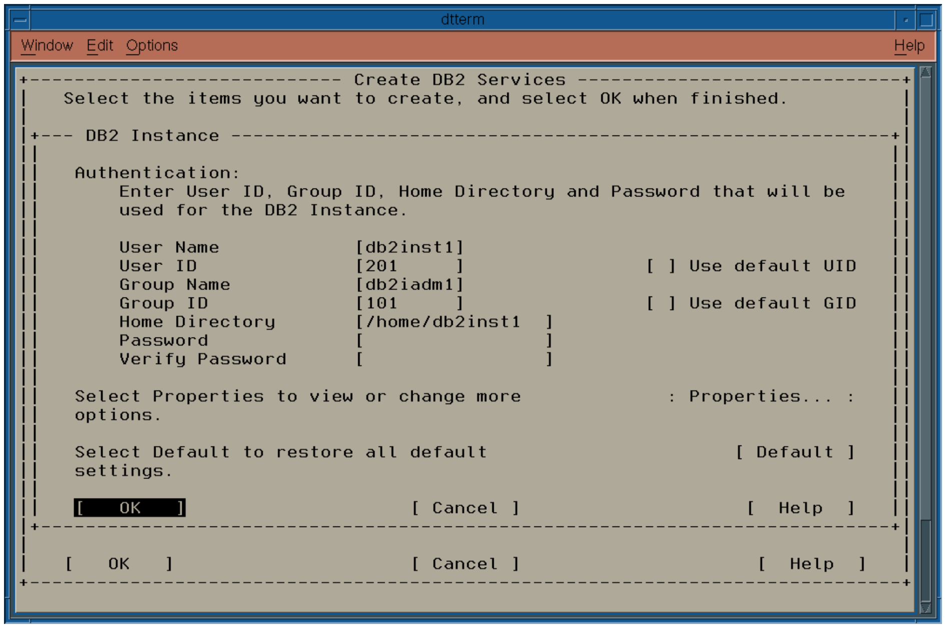
2. As shown in Figure 6-23, select **DB2 Administration Client**, move the curser to OK and press the **Enter** key.



```
+---------------------------- Create DB2 Services ----------------------------+
|   Select the items you want to create, and select OK when finished.         |
|                                                                              |
|   A DB2 Instance is an environment where you store data and run             |
|   applications.  An instance can contain multiple databases.                |
|                                                                              |
|  (*) Create a DB2 Instance.                           [ Customize... ]     |
|   ( ) Do not create a DB2 Instance.                                         |
|                                                                              |
|   [   OK   ]                 [ Cancel ]                  [  Help  ]       |
+------------------------------------------------------------------------------+
```

Figure 6-24 DB2 instance creation window

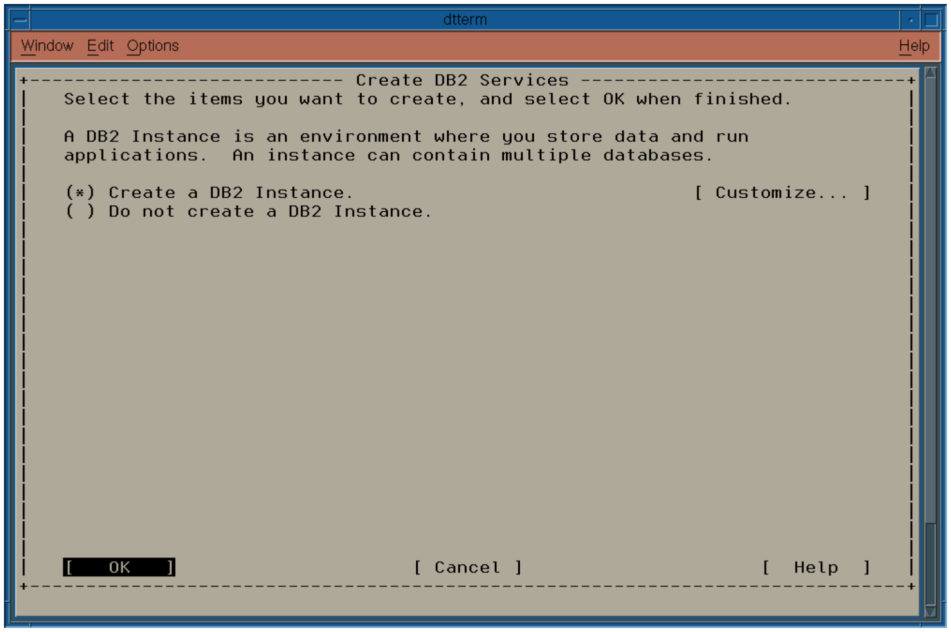
3. On the Create DB2 Services screen, select **Create a DB2 Instance**, move the cursor to OK and press the **Enter** key.



```
+----------------------------- Create DB2 Services -----------------------------+
|    Select the items you want to create, and select OK when finished.          |
|                                                                               |
|+--- DB2 Instance ------------------------------------------------------------+|
||                                                                            ||
||    Authentication:                                                         ||
||        Enter User ID, Group ID, Home Directory and Password that will be   ||
||        used for the DB2 Instance.                                          ||
||                                                                            ||
||        User Name               [db2inst1]                                  ||
||        User ID                 [201     ]         [ ] Use default UID      ||
||        Group Name              [db2iadm1]                                  ||
||        Group ID                [101     ]         [ ] Use default GID      ||
||        Home Directory          [/home/db2inst1  ]                          ||
||        Password                [                ]                          ||
||        Verify Password         [                ]                          ||
||                                                                            ||
||    Select Properties to view or change more           : Properties... :    ||
||    options.                                                                ||
||                                                                            ||
||    Select Default to restore all default                    [ Default ]    ||
||    settings.                                                               ||
||                                                                            ||
||    [   OK   ]                 [ Cancel ]                    [  Help  ]     ||
|+----------------------------------------------------------------------------+|
|                                                                               |
|   [   OK   ]                   [ Cancel ]                     [  Help  ]     |
+-------------------------------------------------------------------------------+
```

Figure 6-25 Instance details

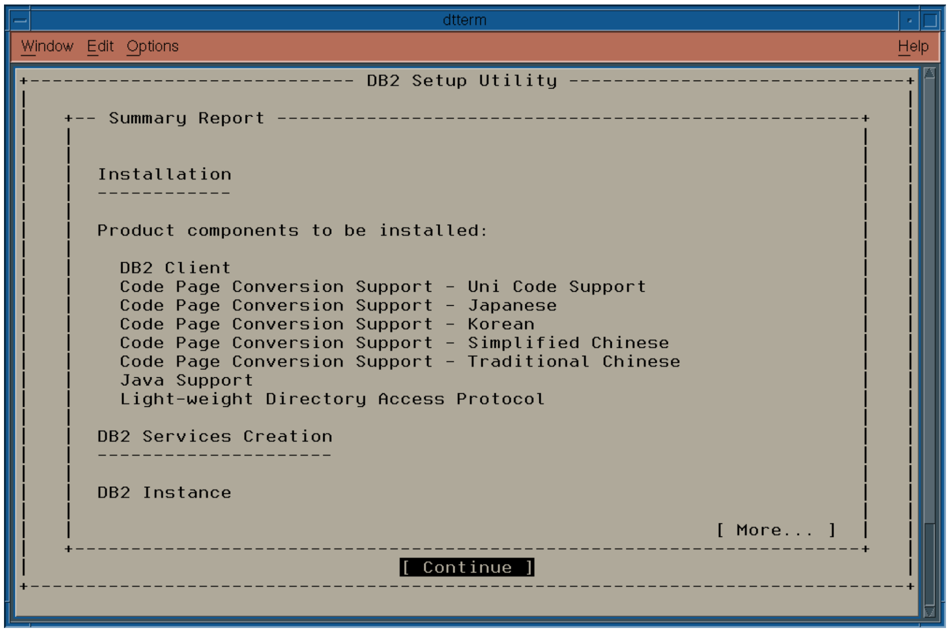
4. The next screen (shown in Figure 6-25) provides default settings for the new instance. If you accept the defaults, the password will be set to ibmdb2. When finished, move the cursor to OK and pressthe **Enter** key.



```
+---------------------------- Create DB2 Services -----------------------------+
|   Select the items you want to create, and select OK when finished.          |
|                                                                              |
|   A DB2 Instance is an environment where you store data and run              |
|   applications.  An instance can contain multiple databases.                 |
|                                                                              |
|   (*) Create a DB2 Instance.                          [ Customize... ]       |
|   ( ) Do not create a DB2 Instance.                                          |
|                                                                              |
|   [    OK    ]               [ Cancel ]                  [  Help  ]         |
+------------------------------------------------------------------------------+
```

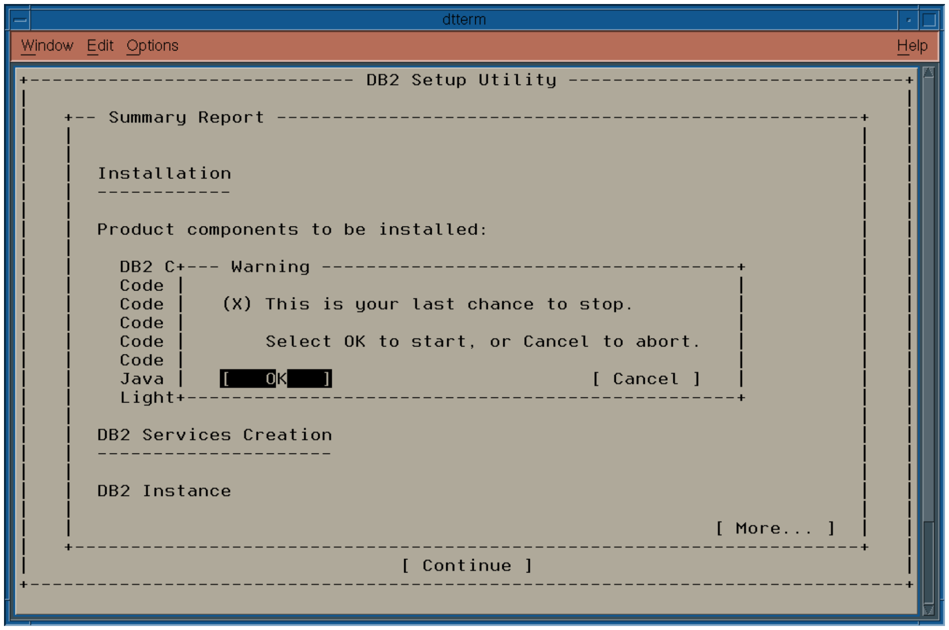
Figure 6-26 Confirm start of installation

5. To start the installation, press **Enter**.
6. The following screens show the progress of the installation and the components that are being installed.



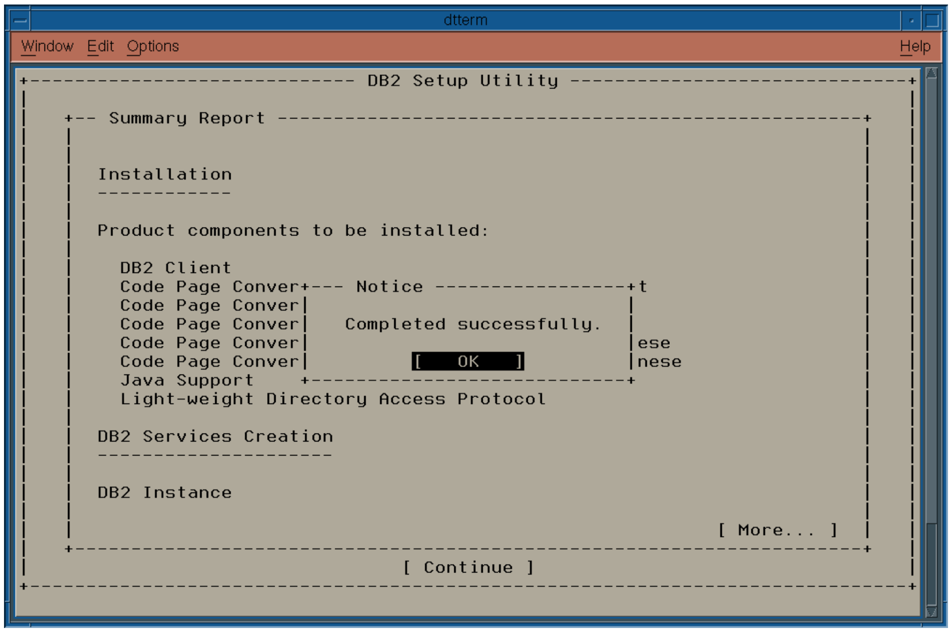
```
+----------------------------- DB2 Setup Utility -------------------------------+
|                                                                               |
|   +-- Summary Report -------------------------------------------------------+ |
|   |                                                                         | |
|   |  Installation                                                           | |
|   |  ------------                                                           | |
|   |                                                                         | |
|   |  Product components to be installed:                                    | |
|   |                                                                         | |
|   |    DB2 Client                                                           | |
|   |    Code Page Conversion Support - Uni Code Support                      | |
|   |    Code Page Conversion Support - Japanese                              | |
|   |    Code Page Conversion Support - Korean                                | |
|   |    Code Page Conversion Support - Simplified Chinese                    | |
|   |    Code Page Conversion Support - Traditional Chinese                   | |
|   |    Java Support                                                         | |
|   |    Light-weight Directory Access Protocol                               | |
|   |                                                                         | |
|   |  DB2 Services Creation                                                  | |
|   |  ---------------------                                                  | |
|   |                                                                         | |
|   |  DB2 Instance                                                           | |
|   |                                                                         | |
|   |                                                          [ More... ]    | |
|   +-------------------------------------------------------------------------+ |
|                                [ Continue ]                                   |
+-------------------------------------------------------------------------------+
```

Figure 6-27 Installing DB2Client



```
+----------------------------- DB2 Setup Utility -------------------------------+
|                                                                               |
|   +-- Summary Report -------------------------------------------------------+ |
|   |                                                                         | |
|   |  Installation                                                           | |
|   |  ------------                                                           | |
|   |                                                                         | |
|   |  Product components to be installed:                                    | |
|   |                                                                         | |
|   |    DB2 C+--- Warning ------------------------------------+               | |
|   |    Code |                                                |              | |
|   |    Code |    (X) This is your last chance to stop.       |              | |
|   |    Code |                                                |              | |
|   |    Code |        Select OK to start, or Cancel to abort. |              | |
|   |    Code |                                                |              | |
|   |    Java |    [    OK    ]                   [ Cancel ]   |              | |
|   |    Light+------------------------------------------------+              | |
|   |                                                                         | |
|   |  DB2 Services Creation                                                  | |
|   |  ---------------------                                                  | |
|   |                                                                         | |
|   |  DB2 Instance                                                           | |
|   |                                                                         | |
|   |                                                          [ More... ]    | |
|   +-------------------------------------------------------------------------+ |
|                                [ Continue ]                                   |
+-------------------------------------------------------------------------------+
```

Figure 6-28 Installing Code page support



```
+------------------------------ DB2 Setup Utility ------------------------------+
|                                                                                |
|   +-- Summary Report ----------------------------------------------------+    |
|   |                                                                      |    |
|   |  Installation                                                        |    |
|   |  ------------                                                        |    |
|   |                                                                      |    |
|   |  Product components to be installed:                                 |    |
|   |                                                                      |    |
|   |    DB2 Client                                                        |    |
|   |    Code Page Conver+--- Notice -----------------+t                   |    |
|   |    Code Page Conver|                            |                    |    |
|   |    Code Page Conver|   Completed successfully.  |                    |    |
|   |    Code Page Conver|                            |ese                 |    |
|   |    Code Page Conver|          [   OK   ]        |nese                |    |
|   |    Java Support    +----------------------------+                    |    |
|   |    Light-weight Directory Access Protocol                            |    |
|   |                                                                      |    |
|   |  DB2 Services Creation                                               |    |
|   |  ---------------------                                               |    |
|   |                                                                      |    |
|   |  DB2 Instance                                                        |    |
|   |                                                                      |    |
|   |                                                        [ More... ]   |    |
|   +----------------------------------------------------------------------+    |
|                               [ Continue ]                                    |
+-------------------------------------------------------------------------------+
```

Figure 6-29 Installation completed successfully

6.1.6 Application Server

We only show the actual installation of WebSphere Application Server on AIX here. For detailed information on other platforms and the preparation tasks involved, see the *WebSphere V4.0 Advanced Edition Handbook;* SG24-6176.

1. Insert the CD into the CD drive on your server.
2. Mount the drive with the following command:

   ```
   mount -rv cdrfs /dev/cd0 /mnt
   ```

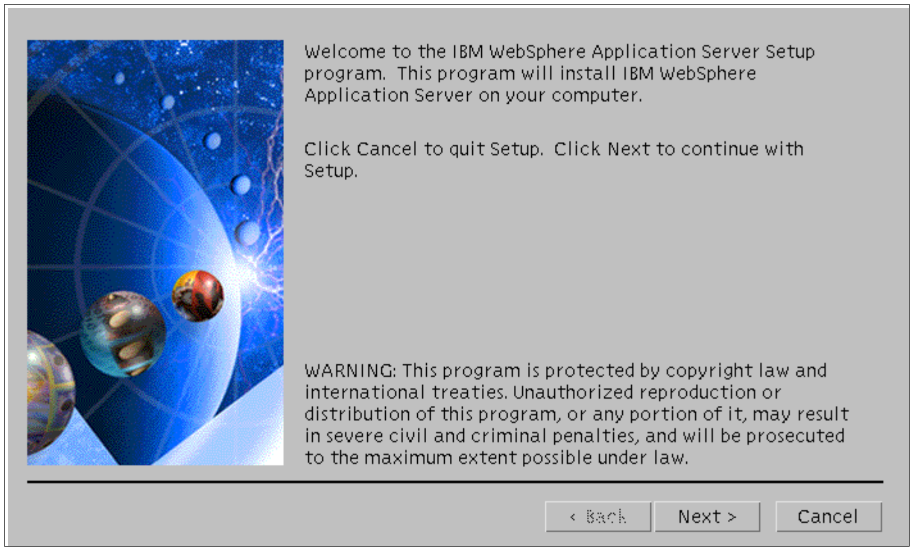
3. Change directory to the AIX directory on the CD-ROM

   ```
   cd /mnt/aix
   ```

4. Start the install script:

   ```
   ./install.sh
   ```

   The WebSphere Application Server welcome panel will be displayed as shown in Figure 6-30.

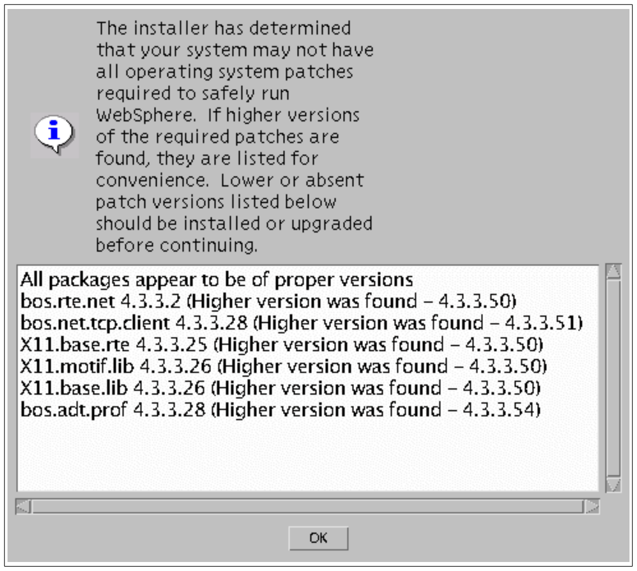

Welcome to the IBM WebSphere Application Server Setup program. This program will install IBM WebSphere Application Server on your computer.

Click Cancel to quit Setup. Click Next to continue with Setup.

WARNING: This program is protected by copyright law and international treaties. Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under law.

Figure 6-30 WebSphere installation welcome window

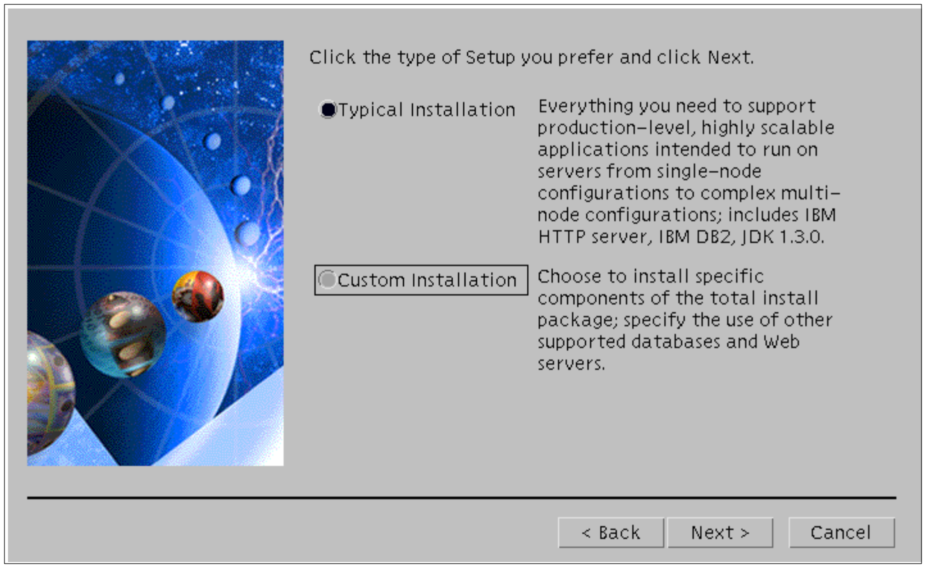
5. Click **Next** to start the installation.


The installer has determined that your system may not have all operating system patches required to safely run WebSphere. If higher versions of the required patches are found, they are listed for convenience. Lower or absent patch versions listed below should be installed or upgraded before continuing.

```
All packages appear to be of proper versions
bos.rte.net 4.3.3.2 (Higher version was found - 4.3.3.50)
bos.net.tcp.client 4.3.3.28 (Higher version was found - 4.3.3.51)
X11.base.rte 4.3.3.25 (Higher version was found - 4.3.3.50)
X11.motif.lib 4.3.3.26 (Higher version was found - 4.3.3.50)
X11.base.lib 4.3.3.26 (Higher version was found - 4.3.3.50)
bos.adt.prof 4.3.3.28 (Higher version was found - 4.3.3.54)
```

Figure 6-31 Prerequisite checker window

6. If your maintenance level does not meet the requirements, you might be prompted by the prereq checker. Install the missing PTFs and start the installation again. If you exceed the prereqs, click **OK** to proceed.

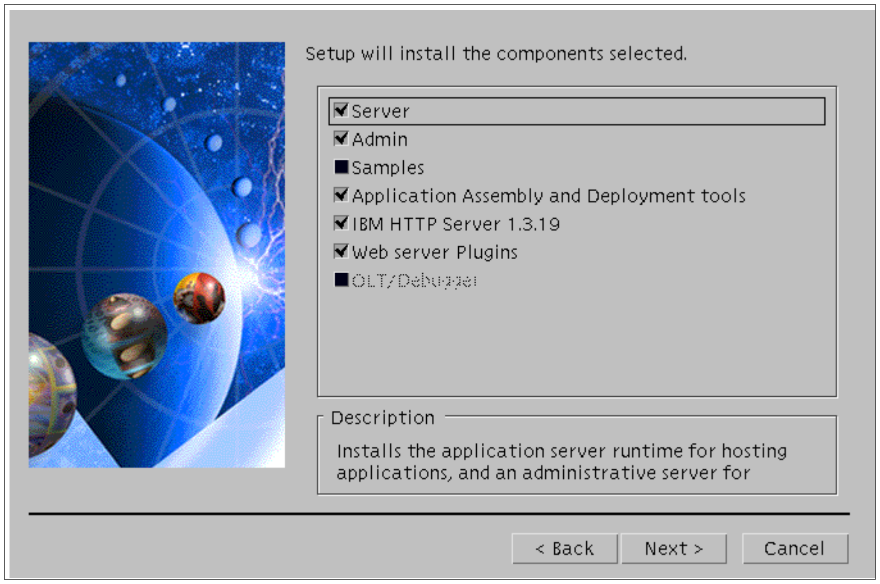

Click the type of Setup you prefer and click Next.

● Typical Installation — Everything you need to support production-level, highly scalable applications intended to run on servers from single-node configurations to complex multi-node configurations; includes IBM HTTP server, IBM DB2, JDK 1.3.0.

○ Custom Installation — Choose to install specific components of the total install package; specify the use of other supported databases and Web servers.

Figure 6-32 Installation type window

7. Select **Custom** for the installation type.


Setup will install the components selected.

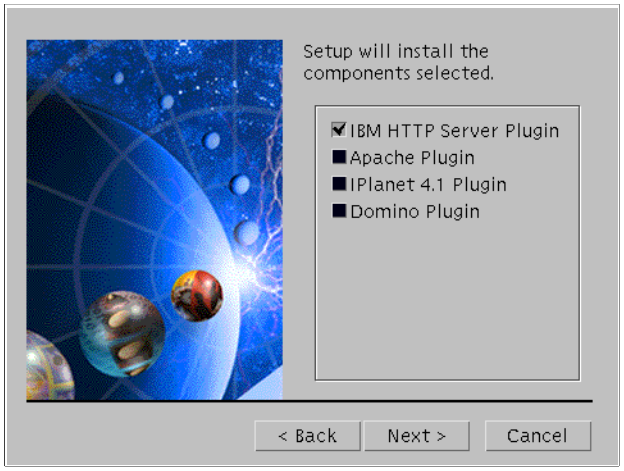
- ✓ Server
- ✓ Admin
- ■ Samples
- ✓ Application Assembly and Deployment tools
- ✓ IBM HTTP Server 1.3.19
- ✓ Web server Plugins
- ■ OLT/Debugger

Description

Installs the application server runtime for hosting applications, and an administrative server for

Figure 6-33 Component selection window

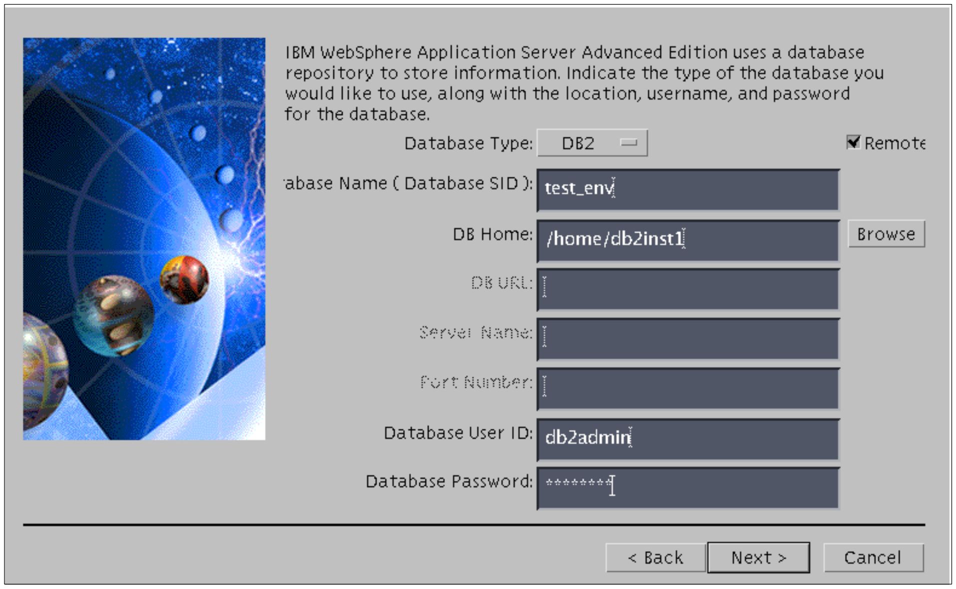
8. Select the following components to be installed:
   – Server
   – Admin
   – Application Assembly and Deployment tools
   – IBM HTTP Server
   – WebServer Plug-ins


Setup will install the components selected.

[✓] IBM HTTP Server Plugin
[ ] Apache Plugin
[ ] IPlanet 4.1 Plugin
[ ] Domino Plugin

Figure 6-34 Plug-in selection window

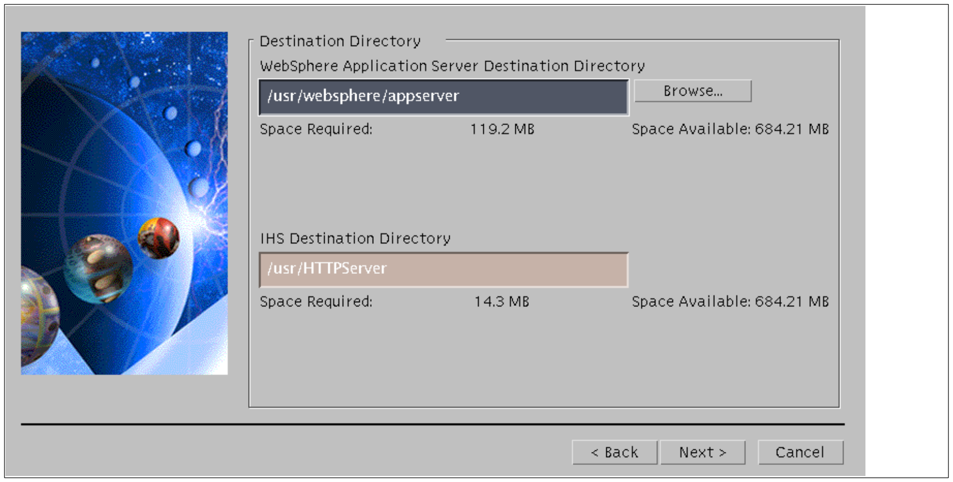
9. As shown in Figure 6-34, we selected the **IBM HTTP Server plug-in**.


IBM WebSphere Application Server Advanced Edition uses a database repository to store information. Indicate the type of the database you would like to use, along with the location, username, and password for the database.

Database Type:	DB2
Remote	✓
abase Name (Database SID):	test_env
DB Home:	/home/db2inst1
DB URL:	
Server Name:	
Port Number:	
Database User ID:	db2admin
Database Password:	********

Figure 6-35 Database details

10. Enter the database details. In our example, they are:
    - Database Type: `DB2`
    - Database name: `test_env`
    - DB Home: `/home/db2inst1`
    - DB User: `db2admin`
    - DB User password: db2admin's password
    - Select the box for **Remote** (Database Server)


Destination Directory

WebSphere Application Server Destination Directory

/usr/websphere/appserver

Space Required: 119.2 MB Space Available: 684.21 MB

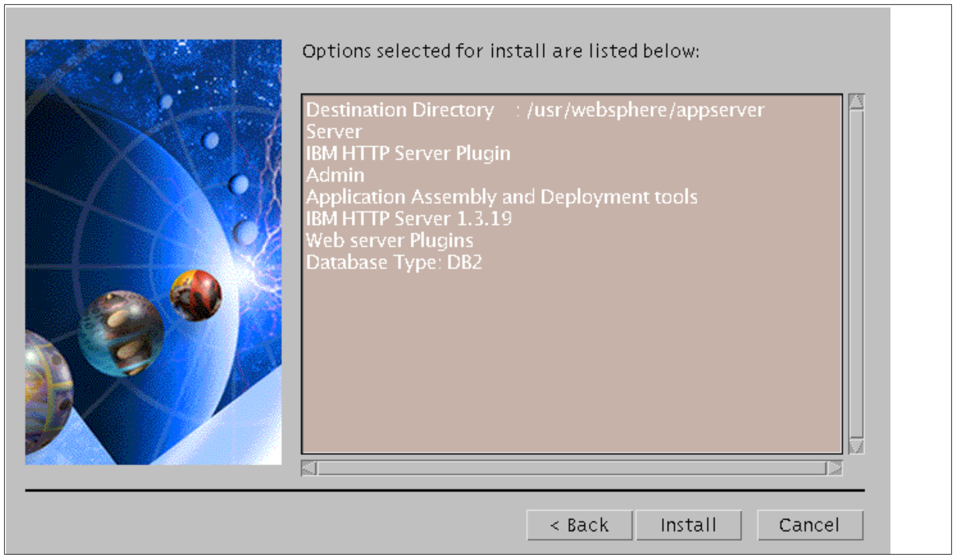
IHS Destination Directory

/usr/HTTPServer

Space Required: 14.3 MB Space Available: 684.21 MB

Figure 6-36 Directory path window

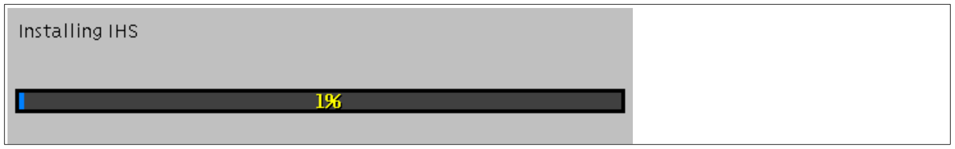
11.Enter the directory path where WebSphere Application Server will be installed. You cannot change the path of IBM HTTP Server.


Options selected for install are listed below:

```
Destination Directory    : /usr/websphere/appserver
Server
IBM HTTP Server Plugin
Admin
Application Assembly and Deployment tools
IBM HTTP Server 1.3.19
Web server Plugins
Database Type: DB2
```

Figure 6-37 Installation summary window

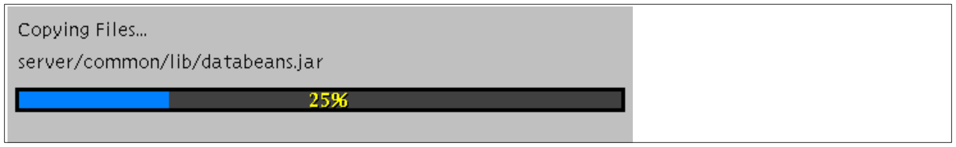
12.The Installation summary window shows the chosen settings. Click **Install** to start the copying of the files.


Installing IHS

1%

Figure 6-38 Component progress window

13. The progress window shows the status of the components installation.

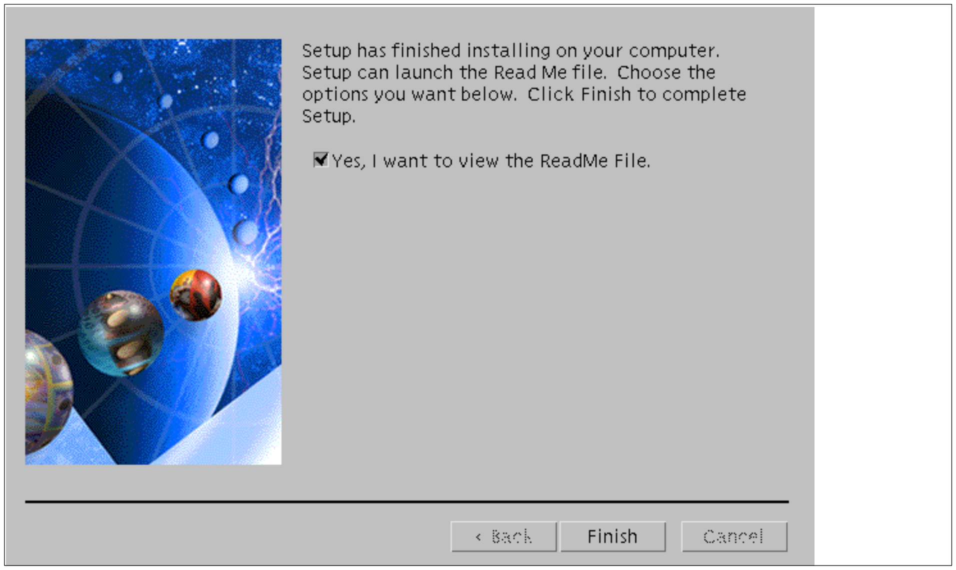

Copying Files...

server/common/lib/databeans.jar

25%

Figure 6-39 File copying progress window

14. After the components installation, files are copied.


Setup has finished installing on your computer. Setup can launch the Read Me file. Choose the options you want below. Click Finish to complete Setup.

☑ Yes, I want to view the ReadMe File.

Figure 6-40 Installation complete window

15. After successful installation, you will be prompted by the Setup Complete window. Click **Finish** to close it.

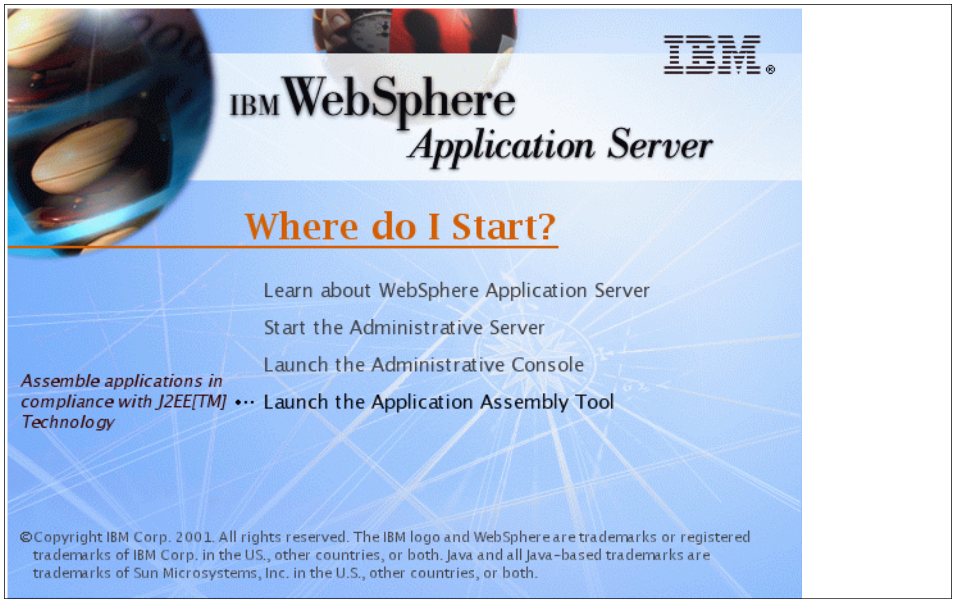
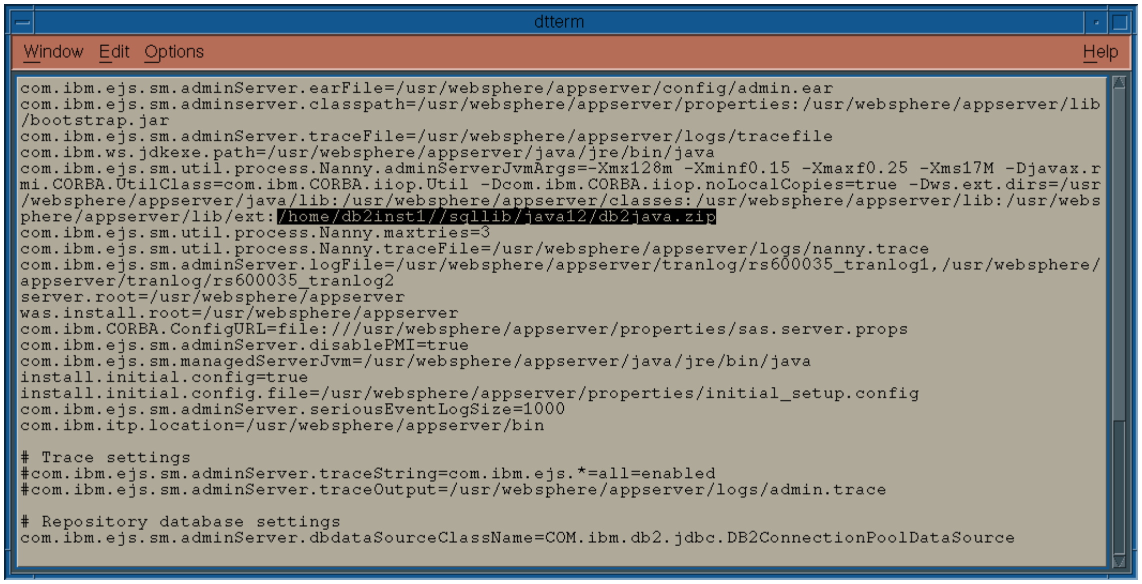

IBM
IBM WebSphere Application Server
Where do I Start?
Learn about WebSphere Application Server
Start the Administrative Server
Launch the Administrative Console
Assemble applications in compliance with J2EE[TM] Technology
Launch the Application Assembly Tool
©Copyright IBM Corp. 2001. All rights reserved. The IBM logo and WebSphere are trademarks or registered trademarks of IBM Corp. in the US., other countries, or both. Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the U.S., other countries, or both.

Figure 6-41 WebSphere auto start window

16. Do not start WebSphere Application Server yet. Close the auto start window.



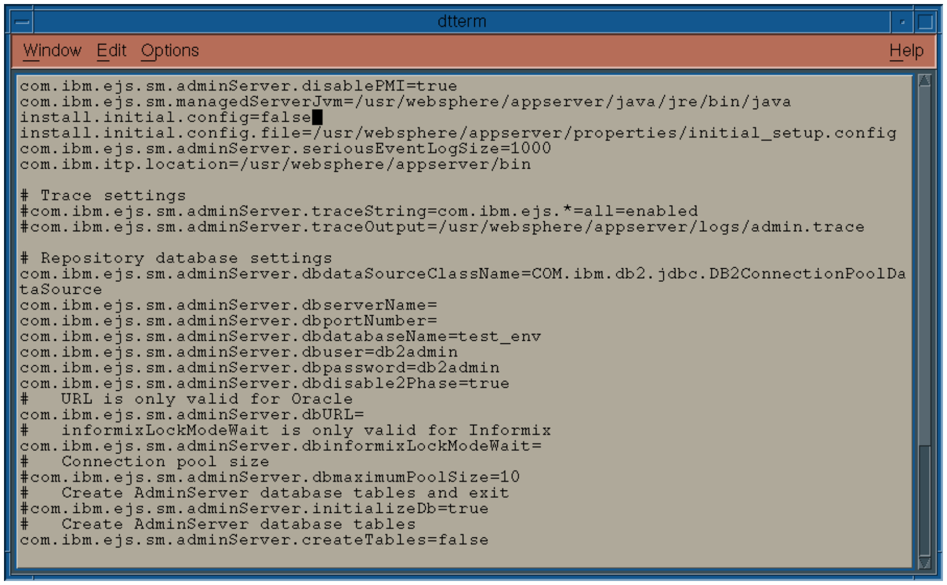
```
com.ibm.ejs.sm.adminServer.earFile=/usr/websphere/appserver/config/admin.ear
com.ibm.ejs.sm.adminserver.classpath=/usr/websphere/appserver/properties:/usr/websphere/appserver/lib
/bootstrap.jar
com.ibm.ejs.sm.adminServer.traceFile=/usr/websphere/appserver/logs/tracefile
com.ibm.ws.jdkexe.path=/usr/websphere/appserver/java/jre/bin/java
com.ibm.ejs.sm.util.process.Nanny.adminServerJvmArgs=-Xmx128m -Xminf0.15 -Xmaxf0.25 -Xms17M -Djavax.r
mi.CORBA.UtilClass=com.ibm.CORBA.iiop.Util -Dcom.ibm.CORBA.iiop.noLocalCopies=true -Dws.ext.dirs=/usr
/websphere/appserver/java/lib:/usr/websphere/appserver/classes:/usr/websphere/appserver/lib:/usr/webs
phere/appserver/lib/ext:/home/db2inst1//sqllib/java12/db2java.zip
com.ibm.ejs.sm.util.process.Nanny.maxtries=3
com.ibm.ejs.sm.util.process.Nanny.traceFile=/usr/websphere/appserver/logs/nanny.trace
com.ibm.ejs.sm.adminServer.logFile=/usr/websphere/appserver/tranlog/rs600035_tranlog1,/usr/websphere/
appserver/tranlog/rs600035_tranlog2
server.root=/usr/websphere/appserver
was.install.root=/usr/websphere/appserver
com.ibm.CORBA.ConfigURL=file:///usr/websphere/appserver/properties/sas.server.props
com.ibm.ejs.sm.adminServer.disablePMI=true
com.ibm.ejs.sm.managedServerJvm=/usr/websphere/appserver/java/jre/bin/java
install.initial.config=true
install.initial.config.file=/usr/websphere/appserver/properties/initial_setup.config
com.ibm.ejs.sm.adminServer.seriousEventLogSize=1000
com.ibm.itp.location=/usr/websphere/appserver/bin

# Trace settings
#com.ibm.ejs.sm.adminServer.traceString=com.ibm.ejs.*=all=enabled
#com.ibm.ejs.sm.adminServer.traceOutput=/usr/websphere/appserver/logs/admin.trace

# Repository database settings
com.ibm.ejs.sm.adminServer.dbdataSourceClassName=COM.ibm.db2.jdbc.DB2ConnectionPoolDataSource
```

Figure 6-42 The admin.config file

17. Go to the WebSphere bin directory. On our installation, the path is:

```
/usr/websphere/appserver/bin
```

View the admin.config file with your favorite editor, or do a `cat` or `more` on it. Make sure that the path to the db2java.zip file points to the java12 directory.



```
com.ibm.ejs.sm.adminServer.disablePMI=true
com.ibm.ejs.sm.managedServerJvm=/usr/websphere/appserver/java/jre/bin/java
install.initial.config=false
install.initial.config.file=/usr/websphere/appserver/properties/initial_setup.config
com.ibm.ejs.sm.adminServer.seriousEventLogSize=1000
com.ibm.itp.location=/usr/websphere/appserver/bin

# Trace settings
#com.ibm.ejs.sm.adminServer.traceString=com.ibm.ejs.*=all=enabled
#com.ibm.ejs.sm.adminServer.traceOutput=/usr/websphere/appserver/logs/admin.trace

# Repository database settings
com.ibm.ejs.sm.adminServer.dbdataSourceClassName=COM.ibm.db2.jdbc.DB2ConnectionPoolDa
taSource
com.ibm.ejs.sm.adminServer.dbserverName=
com.ibm.ejs.sm.adminServer.dbportNumber=
com.ibm.ejs.sm.adminServer.dbdatabaseName=test_env
com.ibm.ejs.sm.adminServer.dbuser=db2admin
com.ibm.ejs.sm.adminServer.dbpassword=db2admin
com.ibm.ejs.sm.adminServer.dbdisable2Phase=true
#   URL is only valid for Oracle
com.ibm.ejs.sm.adminServer.dbURL=
#   informixLockModeWait is only valid for Informix
com.ibm.ejs.sm.adminServer.dbinformixLockModeWait=
#   Connection pool size
#com.ibm.ejs.sm.adminServer.dbmaximumPoolSize=10
#   Create AdminServer database tables and exit
#com.ibm.ejs.sm.adminServer.initializeDb=true
#   Create AdminServer database tables
com.ibm.ejs.sm.adminServer.createTables=false
```

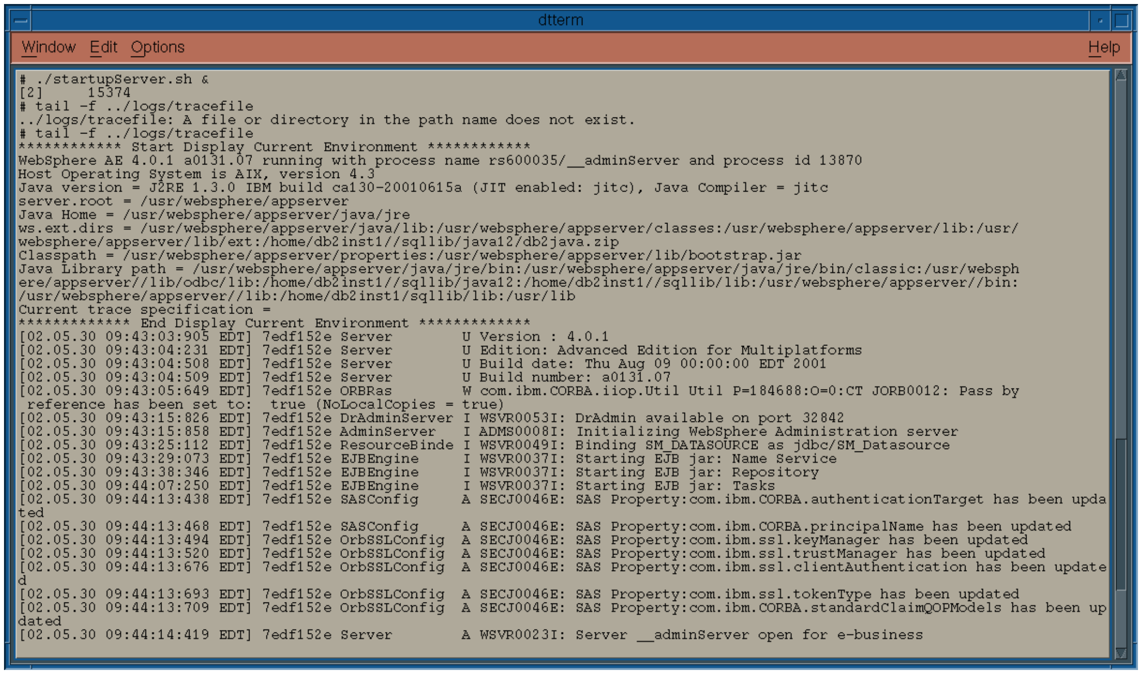
Figure 6-43 The admin.config file

18. If this instance of WebSphere Application Server will connect to a database where you have already performed configurations, change the property that forces WebSphere to recreate the database tables to `false`:

```
com.ibm.ejs.sm.adminServer.createTables=false
```

19. Now start the admin server. If you send it to the background by adding the `&` character, the prompt will return and you will be able to carry on in the same shell:

```
./startupServer.sh &
```



```
# ./startupServer.sh &
[2]     15374
# tail -f ../logs/tracefile
../logs/tracefile: A file or directory in the path name does not exist.
# tail -f ../logs/tracefile
************ Start Display Current Environment ************
WebSphere AE 4.0.1 a0131.07 running with process name rs600035/__adminServer and process id 13870
Host Operating System is AIX, version 4.3
Java version = J2RE 1.3.0 IBM build ca130-20010615a (JIT enabled: jitc), Java Compiler = jitc
server.root = /usr/websphere/appserver
Java Home = /usr/websphere/appserver/java/jre
ws.ext.dirs = /usr/websphere/appserver/java/lib:/usr/websphere/appserver/classes:/usr/websphere/appserver/lib:/usr/
websphere/appserver/lib/ext:/home/db2inst1//sqllib/java12/db2java.zip
Classpath = /usr/websphere/appserver/properties:/usr/websphere/appserver/lib/bootstrap.jar
Java Library path = /usr/websphere/appserver/java/jre/bin:/usr/websphere/appserver/java/jre/bin/classic:/usr/websph
ere/appserver//lib/odbc/lib:/home/db2inst1//sqllib/java12:/home/db2inst1//sqllib/lib:/usr/websphere/appserver//bin:
/usr/websphere/appserver//lib:/home/db2inst1/sqllib/lib:/usr/lib
Current trace specification =
************* End Display Current Environment *************
[02.05.30 09:43:03:905 EDT] 7edf152e Server         U Version : 4.0.1
[02.05.30 09:43:04:231 EDT] 7edf152e Server         U Edition: Advanced Edition for Multiplatforms
[02.05.30 09:43:04:508 EDT] 7edf152e Server         U Build date: Thu Aug 09 00:00:00 EDT 2001
[02.05.30 09:43:04:509 EDT] 7edf152e Server         U Build number: a0131.07
[02.05.30 09:43:05:649 EDT] 7edf152e ORBRas         W com.ibm.CORBA.iiop.Util Util P=184688:O=0:CT JORB0012: Pass by
 reference has been set to:  true (NoLocalCopies = true)
[02.05.30 09:43:15:826 EDT] 7edf152e DrAdminServer I WSVR0053I: DrAdmin available on port 32842
[02.05.30 09:43:15:858 EDT] 7edf152e AdminServer   I ADMS0008I: Initializing WebSphere Administration server
[02.05.30 09:43:25:112 EDT] 7edf152e ResourceBinde I WSVR0049I: Binding SM_DATASOURCE as jdbc/SM_Datasource
[02.05.30 09:43:29:073 EDT] 7edf152e EJBEngine      I WSVR0037I: Starting EJB jar: Name Service
[02.05.30 09:43:38:346 EDT] 7edf152e EJBEngine      I WSVR0037I: Starting EJB jar: Repository
[02.05.30 09:44:07:250 EDT] 7edf152e EJBEngine      I WSVR0037I: Starting EJB jar: Tasks
[02.05.30 09:44:13:438 EDT] 7edf152e SASConfig      A SECJ0046E: SAS Property:com.ibm.CORBA.authenticationTarget has been upda
ted
[02.05.30 09:44:13:468 EDT] 7edf152e SASConfig      A SECJ0046E: SAS Property:com.ibm.CORBA.principalName has been updated
[02.05.30 09:44:13:494 EDT] 7edf152e OrbSSLConfig  A SECJ0046E: SAS Property:com.ibm.ssl.keyManager has been updated
[02.05.30 09:44:13:520 EDT] 7edf152e OrbSSLConfig  A SECJ0046E: SAS Property:com.ibm.ssl.trustManager has been updated
[02.05.30 09:44:13:676 EDT] 7edf152e OrbSSLConfig  A SECJ0046E: SAS Property:com.ibm.ssl.clientAuthentication has been update
d
[02.05.30 09:44:13:693 EDT] 7edf152e OrbSSLConfig  A SECJ0046E: SAS Property:com.ibm.ssl.tokenType has been updated
[02.05.30 09:44:13:709 EDT] 7edf152e OrbSSLConfig  A SECJ0046E: SAS Property:com.ibm.CORBA.standardClaimQOPModels has been up
dated
[02.05.30 09:44:14:419 EDT] 7edf152e Server         A WSVR0023I: Server __adminServer open for e-business
```

Figure 6-44 Tail output of the tracefile

20. Follow the startup of the admin server by looking at the logfile named tracefile:

    ```
    tail -f ../logs/tracefile
    ```

21. Wait until the following message appears.Then exit from tail with **Control + C**:

    ```
    admin-server open for e-business
    ```

6.2 Common configuration guidelines

The following sections document additional configuration requirements to support Load Balancer configurations.

6.2.1 Aliasing a loopback adapter

On Windows

The MS Loopback adapter is not installed by default on Windows. You may have to install it first.

1. Select **Start -> Settings -> Control Panel -> Add/Remove Hardware**.

2. Install the MS Loopback adapter

To configure an alias to the loopback adapter, perform the following steps:

1. Select **Start -> Settings -> Network and Dial Up connections** and select the connection for the loopback adapter.

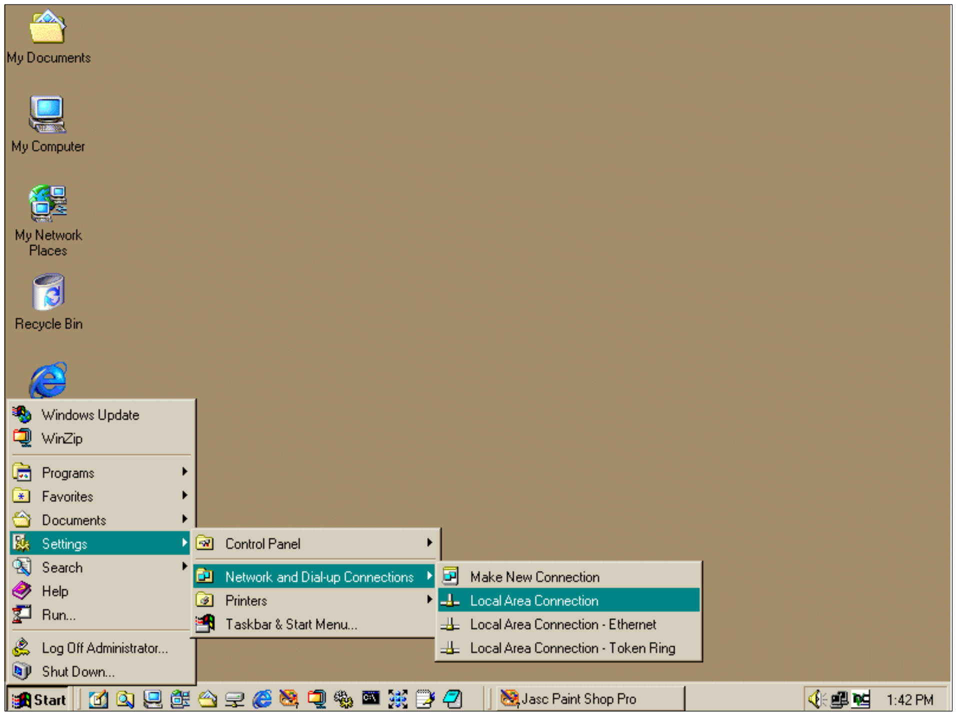
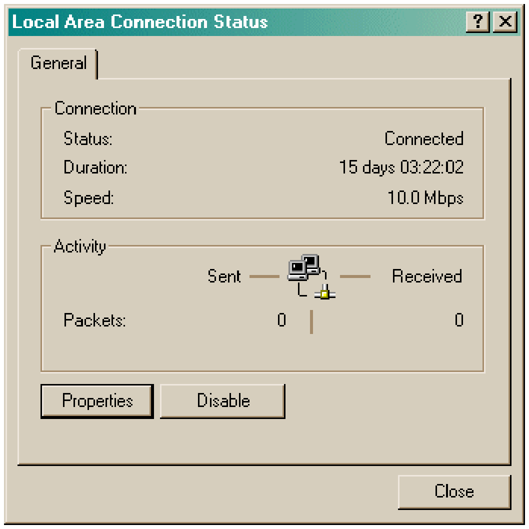


Figure 6-45 Select the Local Area Connection for the loopback adapter



Connection

Status: Connected

Duration: 15 days 03:22:02

Speed: 10.0 Mbps

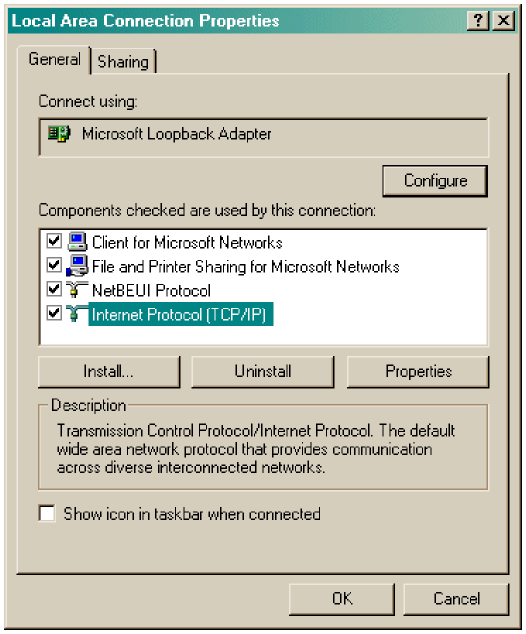
Activity

Sent — — Received

Packets: 0 | 0

Figure 6-46 Local Area Connection Window

2. Click the **Properties** button.



Connect using:

Microsoft Loopback Adapter

Components checked are used by this connection:

☑ Client for Microsoft Networks

☑ File and Printer Sharing for Microsoft Networks

☑ NetBEUI Protocol

☑ Internet Protocol (TCP/IP)

Description

Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

☐ Show icon in taskbar when connected

Figure 6-47 LAN Properties window

3. In the Local Area Network Properties window shown in Figure 6-47, select **Internet Protocol TCP/IP** and click **Properties**.

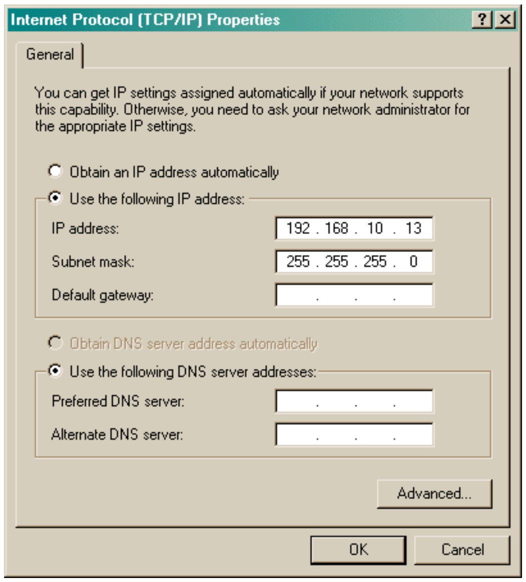
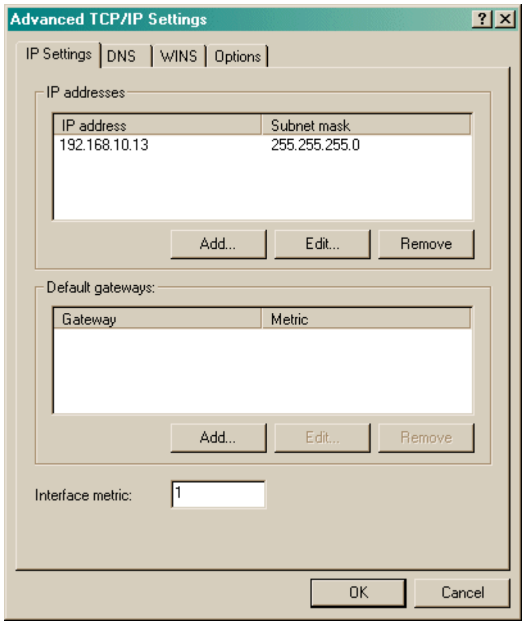

Internet Protocol (TCP/IP) Properties
General
You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.
Obtain an IP address automatically
Use the following IP address:
IP address: 192 . 168 . 10 . 13
Subnet mask: 255 . 255 . 255 . 0
Default gateway:
Obtain DNS server address automatically
Use the following DNS server addresses:
Preferred DNS server:
Alternate DNS server:
Advanced...
OK
Cancel

Figure 6-48 TCP/IP Properties window

4. In the TCP/IP Properties window shown in Figure 6-48, specify the cluster's IP address and the associated subnet mask. If you have already specified another address here, click **Advanced**. Otherwise, click **OK** and proceed with Step 8 on page 170.



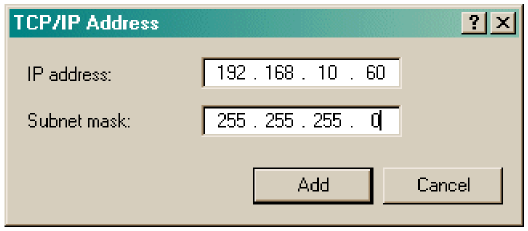
IP address	Subnet mask
192.168.10.13	255.255.255.0

Gateway	Metric

Interface metric: 1

Figure 6-49 Advanced TCP/IP properties

5. In the Advanced TCP/IP properties window, click **Add** in the IP Address area.

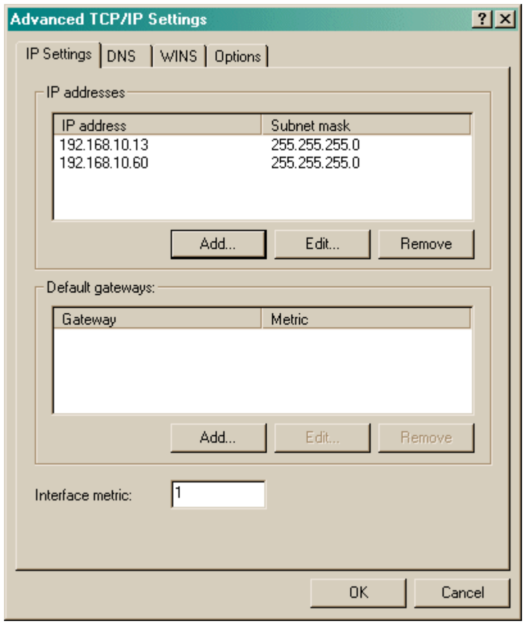


IP address: 192 . 168 . 10 . 60

Subnet mask: 255 . 255 . 255 . 0

Figure 6-50 Add TCP/IP address

6. Enter the IP address of your cluster and the associated subnet mask. Click **OK** to proceed.



IP address	Subnet mask
192.168.10.13	255.255.255.0
192.168.10.60	255.255.255.0

Interface metric: 1

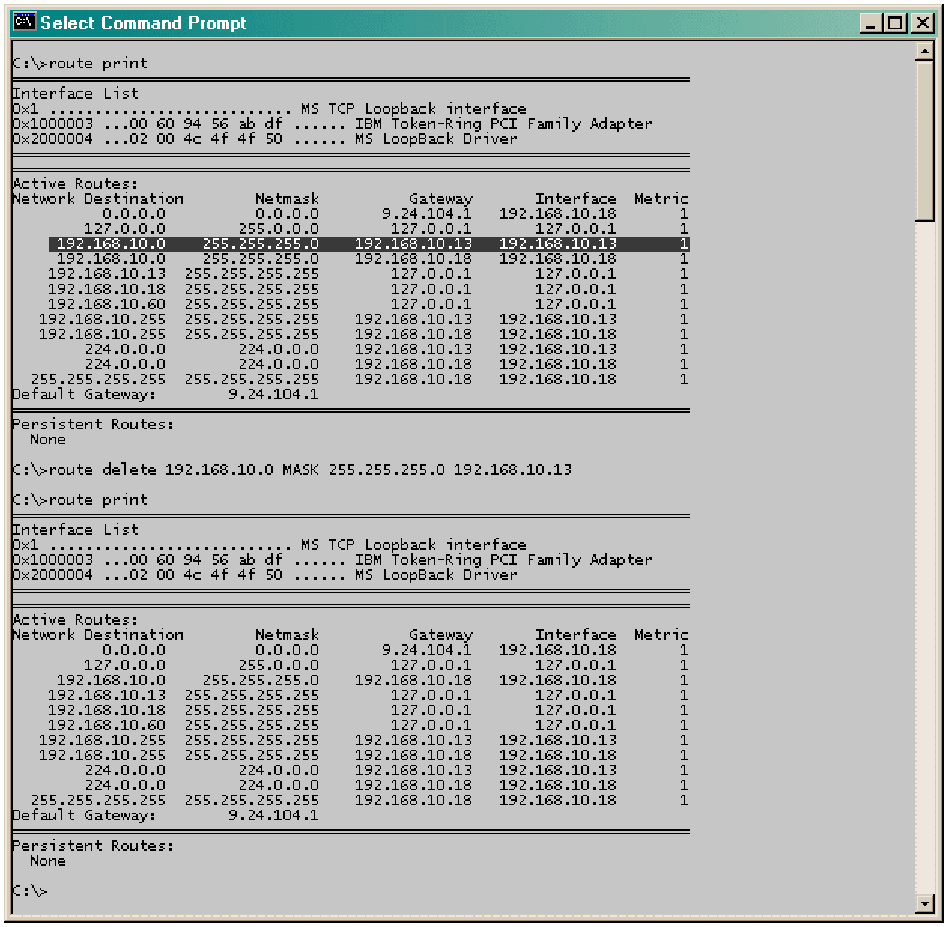
Figure 6-51 Added alias

7. Now the new alias IP address appears in the IP address field. Click **OK** through all the open windows to save the settings.
8. On Windows, an alias to the loopback adapter will produce an incorrect entry in the routing table. The wrong route will point to the local network through the cluster address as an interface. If you have multiple aliases, the first in the list will be used for the incorrect entry. Delete this route with the **`route delete`** command. In our environment, the incorrect route was:

   ```
   192.168.10.0 255.255.255.0 192.168.10.13 192.168.10.13
   ```

   The command to delete the route is:

   ```
   route delete 192.168.10.0 MASK 255.255.255.0 192.168.10.13
   ```



```
C:\>route print
===========================================================================
Interface List
0x1 ........................... MS TCP Loopback interface
0x1000003 ...00 60 94 56 ab df ...... IBM Token-Ring PCI Family Adapter
0x2000004 ...02 00 4c 4f 4f 50 ...... MS LoopBack Driver
===========================================================================
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0      9.24.104.1   192.168.10.18       1
        127.0.0.0        255.0.0.0       127.0.0.1       127.0.0.1       1
     192.168.10.0    255.255.255.0   192.168.10.13   192.168.10.13       1
     192.168.10.0    255.255.255.0   192.168.10.18   192.168.10.18       1
    192.168.10.13  255.255.255.255       127.0.0.1       127.0.0.1       1
    192.168.10.18  255.255.255.255       127.0.0.1       127.0.0.1       1
    192.168.10.60  255.255.255.255       127.0.0.1       127.0.0.1       1
   192.168.10.255  255.255.255.255   192.168.10.13   192.168.10.13       1
   192.168.10.255  255.255.255.255   192.168.10.18   192.168.10.18       1
        224.0.0.0        224.0.0.0   192.168.10.13   192.168.10.13       1
        224.0.0.0        224.0.0.0   192.168.10.18   192.168.10.18       1
  255.255.255.255  255.255.255.255   192.168.10.18   192.168.10.18       1
Default Gateway:        9.24.104.1
===========================================================================
Persistent Routes:
  None

C:\>route delete 192.168.10.0 MASK 255.255.255.0 192.168.10.13

C:\>route print
===========================================================================
Interface List
0x1 ........................... MS TCP Loopback interface
0x1000003 ...00 60 94 56 ab df ...... IBM Token-Ring PCI Family Adapter
0x2000004 ...02 00 4c 4f 4f 50 ...... MS LoopBack Driver
===========================================================================
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0      9.24.104.1   192.168.10.18       1
        127.0.0.0        255.0.0.0       127.0.0.1       127.0.0.1       1
     192.168.10.0    255.255.255.0   192.168.10.18   192.168.10.18       1
    192.168.10.13  255.255.255.255       127.0.0.1       127.0.0.1       1
    192.168.10.18  255.255.255.255       127.0.0.1       127.0.0.1       1
    192.168.10.60  255.255.255.255       127.0.0.1       127.0.0.1       1
   192.168.10.255  255.255.255.255   192.168.10.13   192.168.10.13       1
   192.168.10.255  255.255.255.255   192.168.10.18   192.168.10.18       1
        224.0.0.0        224.0.0.0   192.168.10.13   192.168.10.13       1
        224.0.0.0        224.0.0.0   192.168.10.18   192.168.10.18       1
  255.255.255.255  255.255.255.255   192.168.10.18   192.168.10.18       1
Default Gateway:        9.24.104.1
===========================================================================
Persistent Routes:
  None

C:\>
```

Figure 6-52 Incorrect routes in Windows and their deletion

9. Figure 6-52 shows the incorrect route and the deletion of it.

On AIX

On AIX, adding an alias is done using a simple command, the **`ifconfig`** command.

Syntax:

```
ifconfig <interface> <action> <address>
```

The loopback adapter is named lo0. The command to add an alias of 192.168.10.70 to the loopback adapter is:

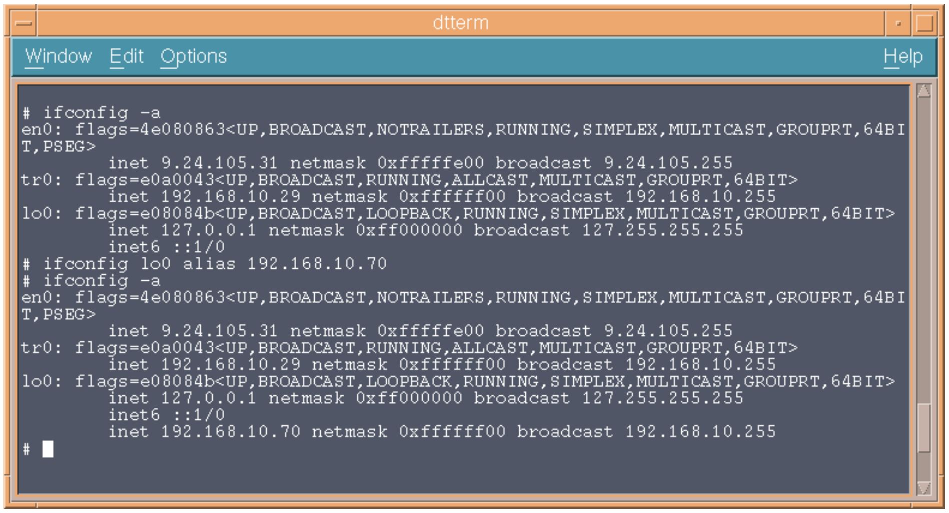
```
ifconfig lo0 alias 192.168.10.70
```

To delete the same alias, you enter:

```
ifconfig lo0 delete 192.168.10.70
```

To show the current settings for all interfaces, enter:

```
ifconfig -a
```



```
# ifconfig -a
en0: flags=4e080863<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64BI
T,PSEG>
        inet 9.24.105.31 netmask 0xfffffe00 broadcast 9.24.105.255
tr0: flags=e0a0043<UP,BROADCAST,RUNNING,ALLCAST,MULTICAST,GROUPRT,64BIT>
        inet 192.168.10.29 netmask 0xffffff00 broadcast 192.168.10.255
lo0: flags=e08084b<UP,BROADCAST,LOOPBACK,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64BIT>
        inet 127.0.0.1 netmask 0xff000000 broadcast 127.255.255.255
        inet6 ::1/0
# ifconfig lo0 alias 192.168.10.70
# ifconfig -a
en0: flags=4e080863<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64BI
T,PSEG>
        inet 9.24.105.31 netmask 0xfffffe00 broadcast 9.24.105.255
tr0: flags=e0a0043<UP,BROADCAST,RUNNING,ALLCAST,MULTICAST,GROUPRT,64BIT>
        inet 192.168.10.29 netmask 0xffffff00 broadcast 192.168.10.255
lo0: flags=e08084b<UP,BROADCAST,LOOPBACK,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64BIT>
        inet 127.0.0.1 netmask 0xff000000 broadcast 127.255.255.255
        inet6 ::1/0
        inet 192.168.10.70 netmask 0xffffff00 broadcast 192.168.10.255
# █
```

Figure 6-53 Ifconfig command on AIX

Figure 6-53 shows the use of the **`ifconfig`** command on AIX.

7

Test environment example

The following example defines a very flexible environment that can be used for application development and testing. Once functional testing on the developer's workstations has been successful, the next stage is to test entire units, also for affinity, multithreading and fail-over matters that require a full scale environment. Because of the way this example is set up, it provides the full range of servers to each developer while still sharing the expensive resources. The developer spaces are fully independent so they will not interfere with each other. An application can be deployed and exported by a developer without the need to understand the whole environment. Those few objects that need to be modified can be changed at deployment time without impacting the original application archive.

7.1 Scenario description

This scenario is based on 2.3.2, “Basic Runtime pattern” on page 29. We deployed the sample Enterprise Application which is delivered with WebSphere Application Server (sampleApp.ear). Each of the above mentioned workspaces contains a ServerGroup with two clones, one virtual host, one database and a DataSource pointing to it.

A single Application Server clone resides on each of the two Application Server nodes. Access goes through two load balanced HTTP servers and the WebSphere plug-ins, which in turn balance between the Application Server Clones (see Figure 7-1).

The virtual host configured for each workspace allows deployment of applications with the same context root for each of them without interfering with others.

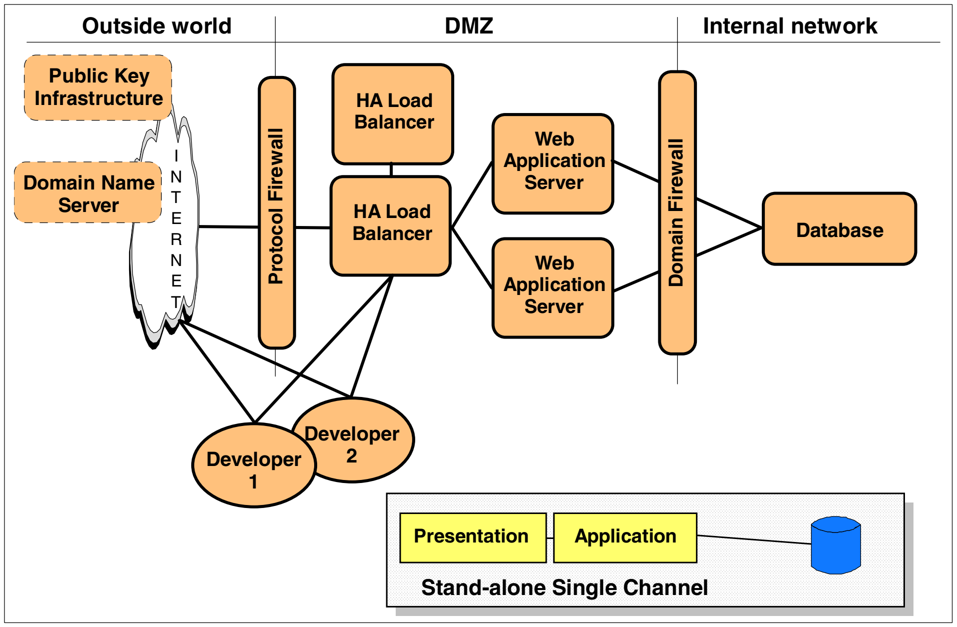


Figure 7-1 Runtime topology of the example test environment

Figure 7-2 shows the architecture of the environment.

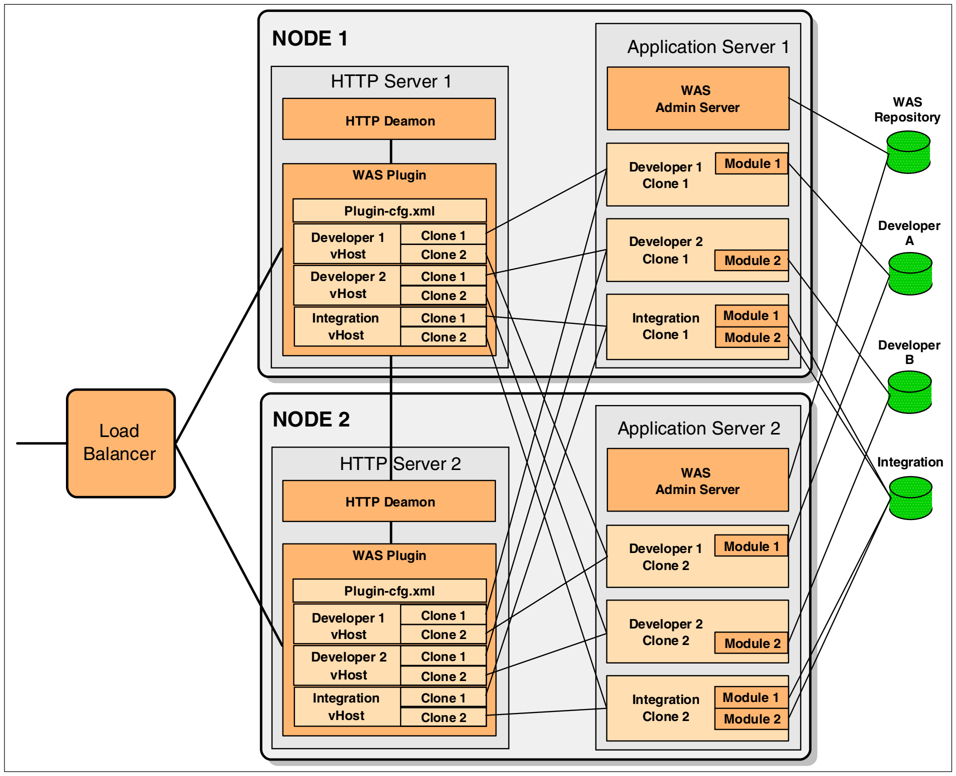


Figure 7-2 Detailed architecture

7.1.1 Overview

We will configure the environment for three different workspaces. Each has its own virtual host, application database and DataSource Object. The following tables show the naming and node functionalities.

Naming

Note that the virtual hosts are name-based virtual hosts. They are all based on a single IP address. This makes it very easy to extend. There is no need for an additional IP address when a new workspace is defined. And even more important, the load balancer configuration never needs to be changed. Simply add a new definition to DNS or the hosts files.

Table 7-1 Naming of user specific configuration items

User	Virtual Host	Database	Data Source	IP Address
int	int_vhost	int_db	int_DS	192.168.10.60
dev1	dev1_vhost	dev1_db	dev1_DS	192.168.10.60
dev2	dev2_vhost	dev2_db	dev2_DS	192.168.10.60

Node functionalities

The following table lists the specifics for the machines used in this scenario. See Figure 7-3 for additional information.

Table 7-2 Machines used in the test environment scenario

Host Name	IP Address	OS and Software	Service
23VNX85	192.168.10.14	Windows 2000 WebSphere Edge Server V2.0	Load Balancer
RS60008	192.168.10.27	AIX 4.3.3 IBM WebSphere Application Server V4.0.3 IBM HTTP Server V 1.3.19	Web Application Server
RS600035	192.168.10.29	AIX 4.3.3 IBM WebSphere Application Server V4.0.3 IBM HTTP Server V 1.3.19	Web Application Server
M23VNX64	9.24.105.112	Windows 2000 IBM DB2 V7.2	Database Server

The entire scenario can be configured with four nodes:

- One load balancer
- Two Web Application Servers
- One Database Server node

For a testing environment, a standalone (not highly available) load balancer is perfectly acceptable. The databases could probably also be created on an existing database server.

The cluster address used for load balancing is 192.168.10.50.

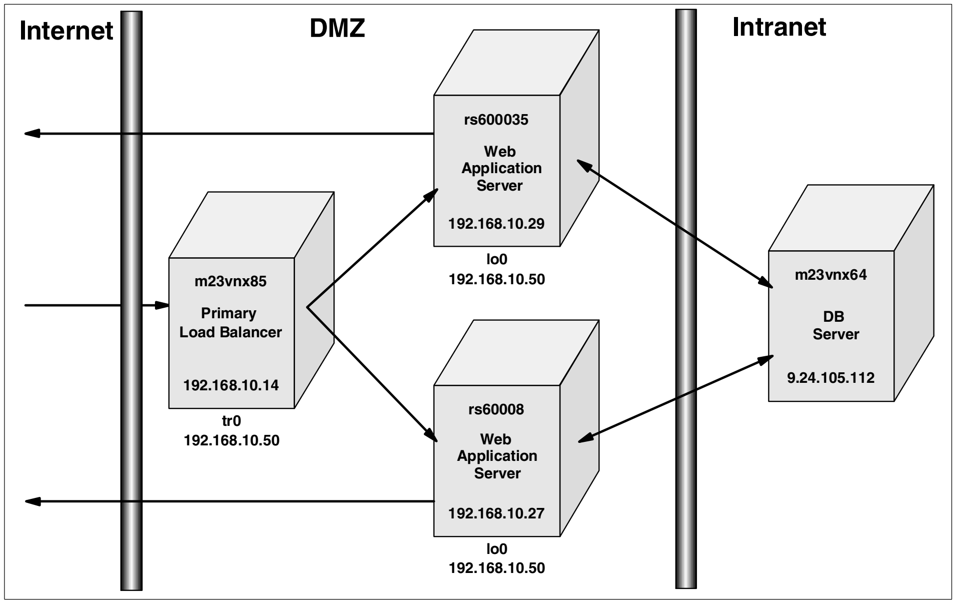


Figure 7-3 Node functionalities

7.1.2 Traffic flow

We will follow a request for a page on one of the virtual hosts, for example:

```
http://dev1_vhost/webapp/examples/showCfg
```

Please refer to Figure 7-4 for details.

1. The hostname in a client request is resolved to the IP address 192.168.10.50. This cluster address is aliased to the interface of the load balancer machine, so the load balancer will answer to the ARP broadcast and receive the request.
2. The load balancer then looks in its connection table to determine the least loaded server in the cluster. In our scenario, we use the MAC forwarding method. It does not read the header of the request. It changes the MAC address of the packet to the one of the target servers and sends the packet on its way. The source and destination IP addresses remain unchanged.
3. The owner of this MAC address is one of our Web Application Servers. But in order to accept the packet, it has to own the IP address as well. How does this work without producing an IP address conflict? The address is aliased to the loopback adapter, which does not respond to ARP broadcasts.

4. The HTTP daemon on the Web Application Server receives the request and lets the WebSphere plug-in examine the header. If it finds a match with a hostname/URI combination in its configuration file, and there is one or more clones (JVM) associated with it, the plug-in chooses one.
5. Each clone in turn has its own HTTP daemon which receives the packet. The application root determines which of the Web Containers served by this daemon owns the URI.
6. The application root /webapp/examples is owned by the Web Container Examples. This Web Container receives the request and forwards it to the Servlet Engine.
7. Finally, the Servlet Engine calls the required servlet or JSP to execute the request. In our example, it is the showCfg servlet.
8. The response is sent back to the WebSphere plug-in. It remembers the original source IP address and sends the response directly to the client.

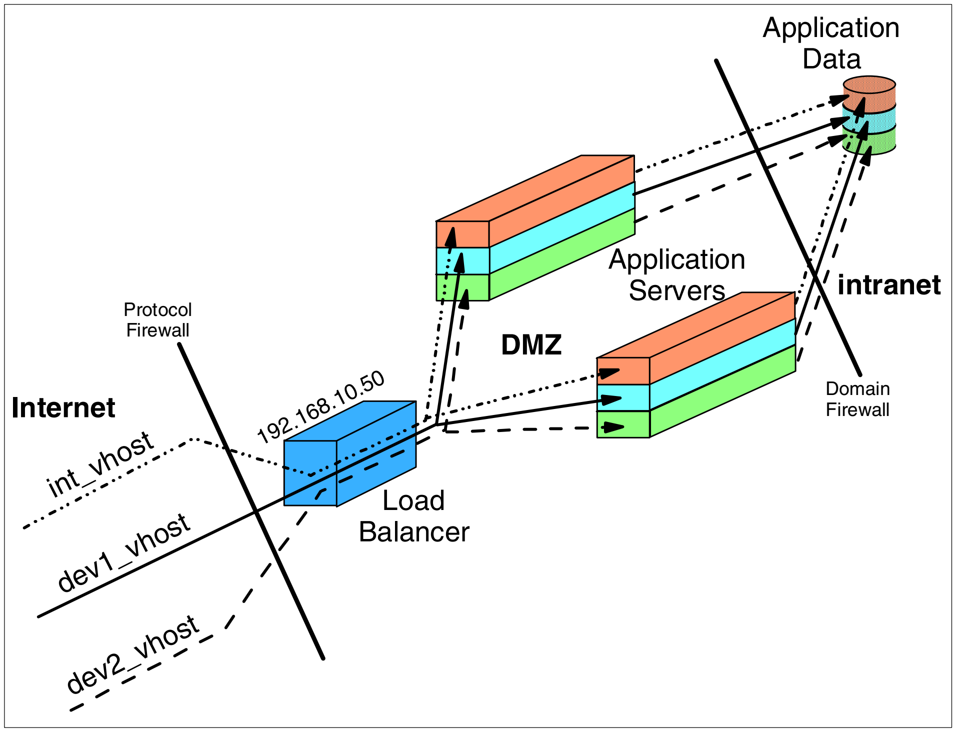


Figure 7-4 Traffic flow

7.2 Configuration description

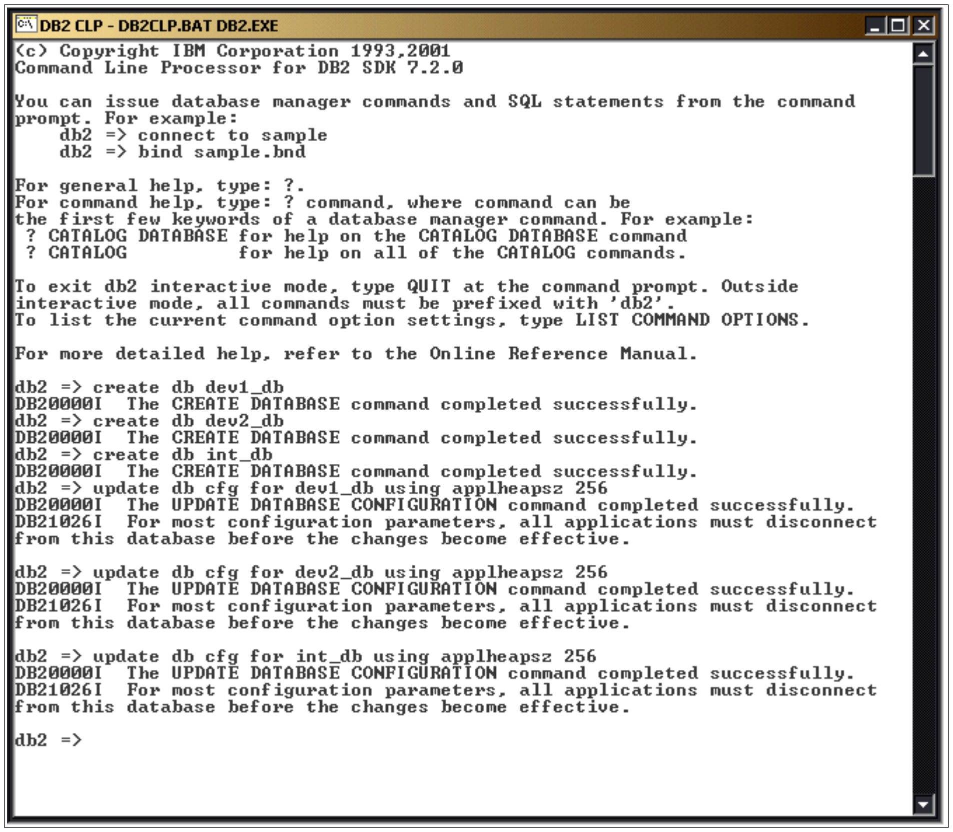
The following sections detail the configuration requirements for the software in each of the nodes in this scenario.

7.2.1 Databases

The instructions in this section are DB2 specific. If you work with a different database, consult your product documentation.

Create a database for each developer

1. On the database server node, log on as the database administrator. Recall that our database server resides on a Windows 2000 machine. The DB administrator is db2admin. Open a command line prompt window by clicking **Start -> Programs -> IBM DB2 -> Command Line Processor**.



```
(c) Copyright IBM Corporation 1993,2001
Command Line Processor for DB2 SDK 7.2.0

You can issue database manager commands and SQL statements from the command
prompt. For example:
    db2 => connect to sample
    db2 => bind sample.bnd

For general help, type: ?.
For command help, type: ? command, where command can be
the first few keywords of a database manager command. For example:
 ? CATALOG DATABASE for help on the CATALOG DATABASE command
 ? CATALOG          for help on all of the CATALOG commands.

To exit db2 interactive mode, type QUIT at the command prompt. Outside
interactive mode, all commands must be prefixed with 'db2'.
To list the current command option settings, type LIST COMMAND OPTIONS.

For more detailed help, refer to the Online Reference Manual.

db2 => create db dev1_db
DB20000I  The CREATE DATABASE command completed successfully.
db2 => create db dev2_db
DB20000I  The CREATE DATABASE command completed successfully.
db2 => create db int_db
DB20000I  The CREATE DATABASE command completed successfully.
db2 => update db cfg for dev1_db using applheapsz 256
DB20000I  The UPDATE DATABASE CONFIGURATION command completed successfully.
DB21026I  For most configuration parameters, all applications must disconnect
from this database before the changes become effective.

db2 => update db cfg for dev2_db using applheapsz 256
DB20000I  The UPDATE DATABASE CONFIGURATION command completed successfully.
DB21026I  For most configuration parameters, all applications must disconnect
from this database before the changes become effective.

db2 => update db cfg for int_db using applheapsz 256
DB20000I  The UPDATE DATABASE CONFIGURATION command completed successfully.
DB21026I  For most configuration parameters, all applications must disconnect
from this database before the changes become effective.

db2 =>
```

Figure 7-5 Creating the databases

2. Create a database for each developer with the following command:

   ```
   create db <dbname>
   ```

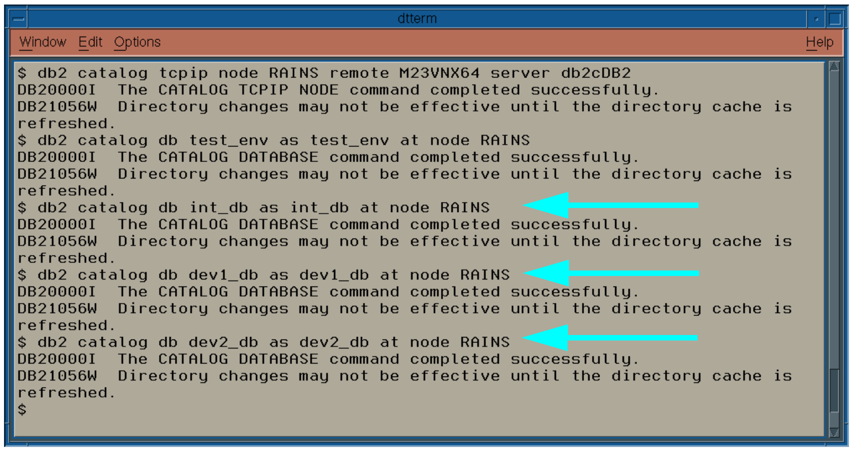
   In the example shown in Figure 7-5, we created databases named dev1_db, dev2_db and int_db.

3. After the databases have been created, set the application heap size to 256 MB with this command:

   ```
   update db cfg for <dbname> using applheapsz 256
   ```

Catalog the new databases on each DB Client node

The following commands must be executed on the respective Web Application Server machines.



```
$ db2 catalog tcpip node RAINS remote M23VNX64 server db2cDB2
DB20000I  The CATALOG TCPIP NODE command completed successfully.
DB21056W  Directory changes may not be effective until the directory cache is
refreshed.
$ db2 catalog db test_env as test_env at node RAINS
DB20000I  The CATALOG DATABASE command completed successfully.
DB21056W  Directory changes may not be effective until the directory cache is
refreshed.
$ db2 catalog db int_db as int_db at node RAINS
DB20000I  The CATALOG DATABASE command completed successfully.
DB21056W  Directory changes may not be effective until the directory cache is
refreshed.
$ db2 catalog db dev1_db as dev1_db at node RAINS
DB20000I  The CATALOG DATABASE command completed successfully.
DB21056W  Directory changes may not be effective until the directory cache is
refreshed.
$ db2 catalog db dev2_db as dev2_db at node RAINS
DB20000I  The CATALOG DATABASE command completed successfully.
DB21056W  Directory changes may not be effective until the directory cache is
refreshed.
$
```

Figure 7-6 Cataloguing the node and databases

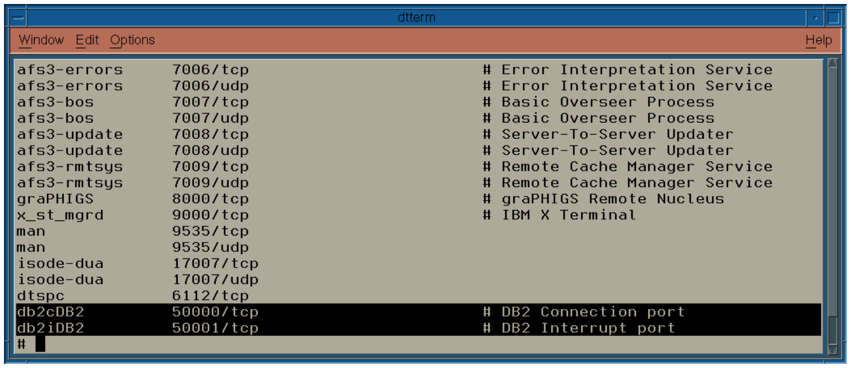
Add node and database catalog entries. The syntax of the catalog command is:

```
catalog db <DB Name> as <DB Alias> at node <DB Server Name>
```

For example, to catalog the dev2_db database, the command would look like this:

```
catalog db dev2_db as dev2_db at node rains
```

See Figure 7-6 for details.



```
afs3-errors     7006/tcp                        # Error Interpretation Service
afs3-errors     7006/udp                        # Error Interpretation Service
afs3-bos        7007/tcp                        # Basic Overseer Process
afs3-bos        7007/udp                        # Basic Overseer Process
afs3-update     7008/tcp                        # Server-To-Server Updater
afs3-update     7008/udp                        # Server-To-Server Updater
afs3-rmtsys     7009/tcp                        # Remote Cache Manager Service
afs3-rmtsys     7009/udp                        # Remote Cache Manager Service
graPHIGS        8000/tcp                        # graPHIGS Remote Nucleus
x_st_mgrd       9000/tcp                        # IBM X Terminal
man             9535/tcp
man             9535/udp
isode-dua       17007/tcp
isode-dua       17007/udp
dtspc           6112/tcp
db2cDB2         50000/tcp                       # DB2 Connection port
db2iDB2         50001/tcp                       # DB2 Interrupt port
#
```

Figure 7-7 Edit the /etc/services file

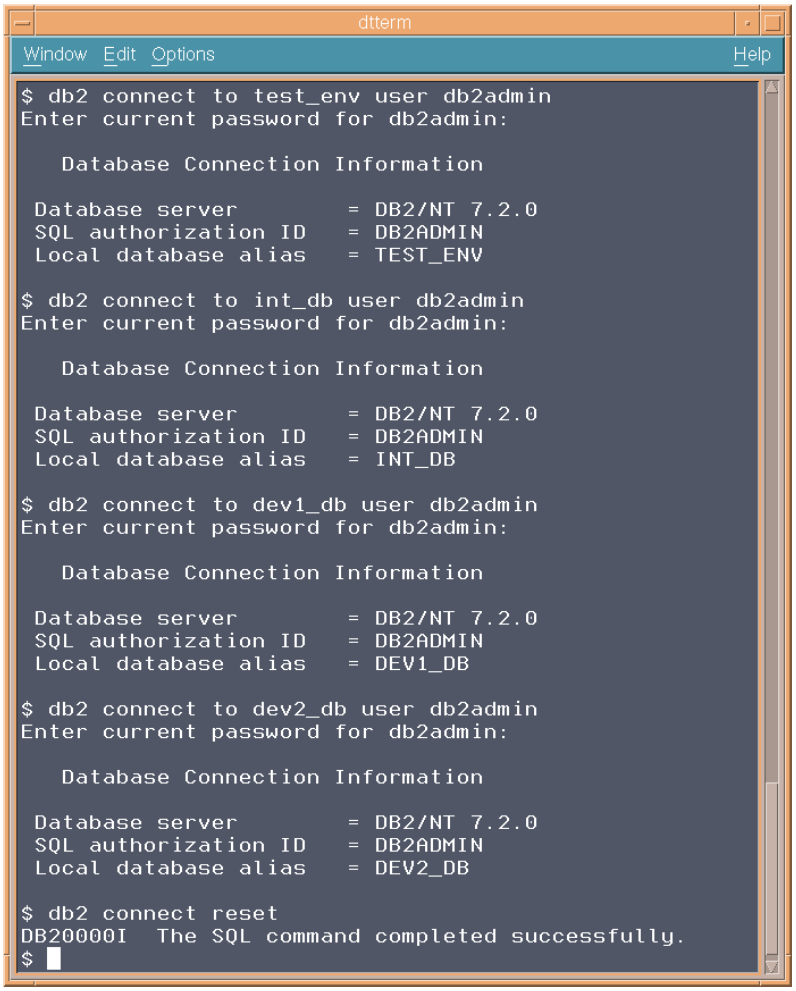
Add two entries to the services file in the /etc directory:

- db2cDB2 50000/tcp

  db2cDB2 is the services name which is bound to a remote host when it is cataloged. The letter *c* in the middle indicates that this is a connection port. *DB2* after it indicates the instance name. If the client has a catalog entry to a Unix DB2 Server, there would probably be another entry with the name db2cDB2inst1.

- db2iDB2 50001/tcp

  This entry defines the interrupt port (indicated by the letter *i* in the middle). *DB2* after it is for the instance name. If the client had a catalog entry to a Unix DB2 Server, there would probably be another entry with the name db2iDB2inst1.



```
$ db2 connect to test_env user db2admin
Enter current password for db2admin:

   Database Connection Information

 Database server        = DB2/NT 7.2.0
 SQL authorization ID   = DB2ADMIN
 Local database alias   = TEST_ENV

$ db2 connect to int_db user db2admin
Enter current password for db2admin:

   Database Connection Information

 Database server        = DB2/NT 7.2.0
 SQL authorization ID   = DB2ADMIN
 Local database alias   = INT_DB

$ db2 connect to dev1_db user db2admin
Enter current password for db2admin:

   Database Connection Information

 Database server        = DB2/NT 7.2.0
 SQL authorization ID   = DB2ADMIN
 Local database alias   = DEV1_DB

$ db2 connect to dev2_db user db2admin
Enter current password for db2admin:

   Database Connection Information

 Database server        = DB2/NT 7.2.0
 SQL authorization ID   = DB2ADMIN
 Local database alias   = DEV2_DB

$ db2 connect reset
DB20000I  The SQL command completed successfully.
$
```

Figure 7-8 Testing the connections

Log on as the local DB2 instance owner (or any user that has the DB2 binaries in its PATH) and verify the connection to the remote Database Server, as shown in Figure 7-8.

Create separate INC Table for each SampleApp

To create the INC table used by the Increment EJB, log on to the application database:

```
db2 connect to int_db user db2admin
```

Once you are connected, issue the following SQL statement to create the table:

```
CREATE TABLE DB2ADMIN.INC ("PRIMARYKEY" VARCHAR (64) NOT NULL, "THEVALUE"
INTEGER, PRIMARY KEY (PRIMARYKEY)) DATA CAPTURE NONE   IN USERSPACE1
```

Repeat this for each developer's database.

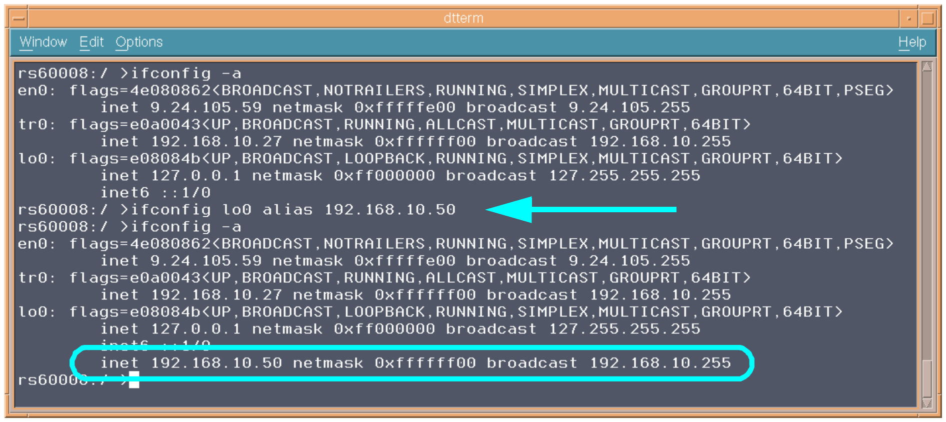
7.2.2 WebSphere Application Server

If you are accessing the WebSphere nodes remotely, export the DISPLAY variable with the IP address of your local machine:

```
export DISPLAY=192.168.10.105:0
```

Alias the loopback adapter

The first step in preparing a member of a cluster is to alias the cluster address to the content server's loopback adapter. In our example, the cluster address is 192.168.10.50. See 6.2.1, "Aliasing a loopback adapter" on page 165 for details.



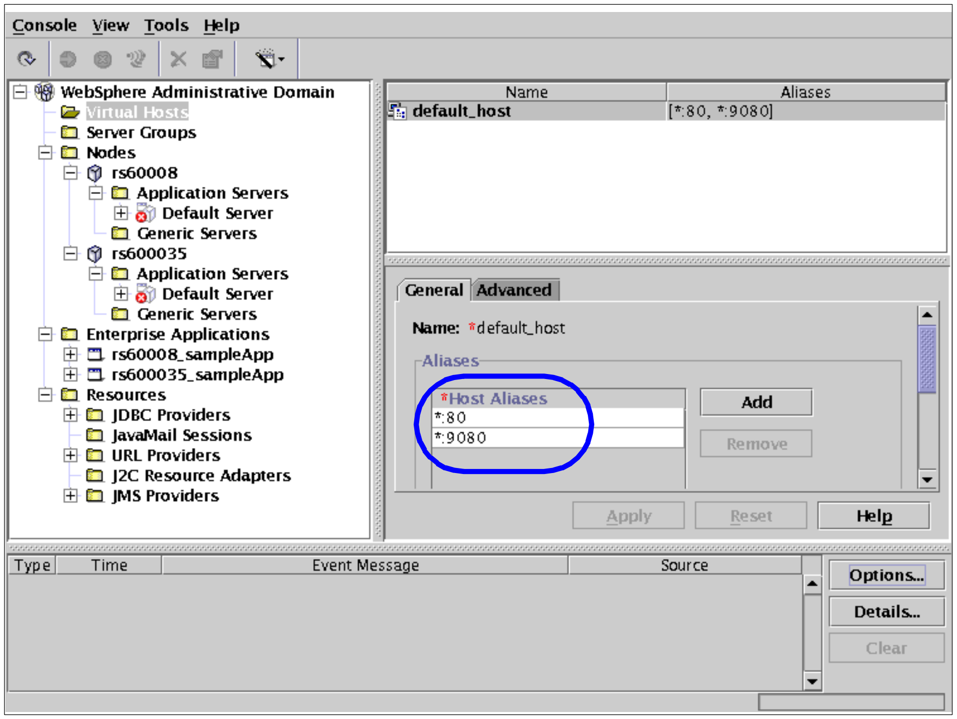
```
rs60008:/ >ifconfig -a
en0: flags=4e080862<BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64BIT,PSEG>
        inet 9.24.105.59 netmask 0xfffffe00 broadcast 9.24.105.255
tr0: flags=e0a0043<UP,BROADCAST,RUNNING,ALLCAST,MULTICAST,GROUPRT,64BIT>
        inet 192.168.10.27 netmask 0xffffff00 broadcast 192.168.10.255
lo0: flags=e08084b<UP,BROADCAST,LOOPBACK,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64BIT>
        inet 127.0.0.1 netmask 0xff000000 broadcast 127.255.255.255
        inet6 ::1/0
rs60008:/ >ifconfig lo0 alias 192.168.10.50
rs60008:/ >ifconfig -a
en0: flags=4e080862<BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64BIT,PSEG>
        inet 9.24.105.59 netmask 0xfffffe00 broadcast 9.24.105.255
tr0: flags=e0a0043<UP,BROADCAST,RUNNING,ALLCAST,MULTICAST,GROUPRT,64BIT>
        inet 192.168.10.27 netmask 0xffffff00 broadcast 192.168.10.255
lo0: flags=e08084b<UP,BROADCAST,LOOPBACK,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64BIT>
        inet 127.0.0.1 netmask 0xff000000 broadcast 127.255.255.255
        inet6 ::1/0
        inet 192.168.10.50 netmask 0xffffff00 broadcast 192.168.10.255
rs60008:/ >
```

Figure 7-9 Aliasing the loopback interface

Configure the virtual hosts in the WebSphere domain

1. Change to WebSphere's binary directory and start the Admin Console.

```
cd /usr/websphere/appserver/bin
./adminclient.sh &
```



Name	Aliases
default_host	[*:80, *:9080]

Name: *default_host

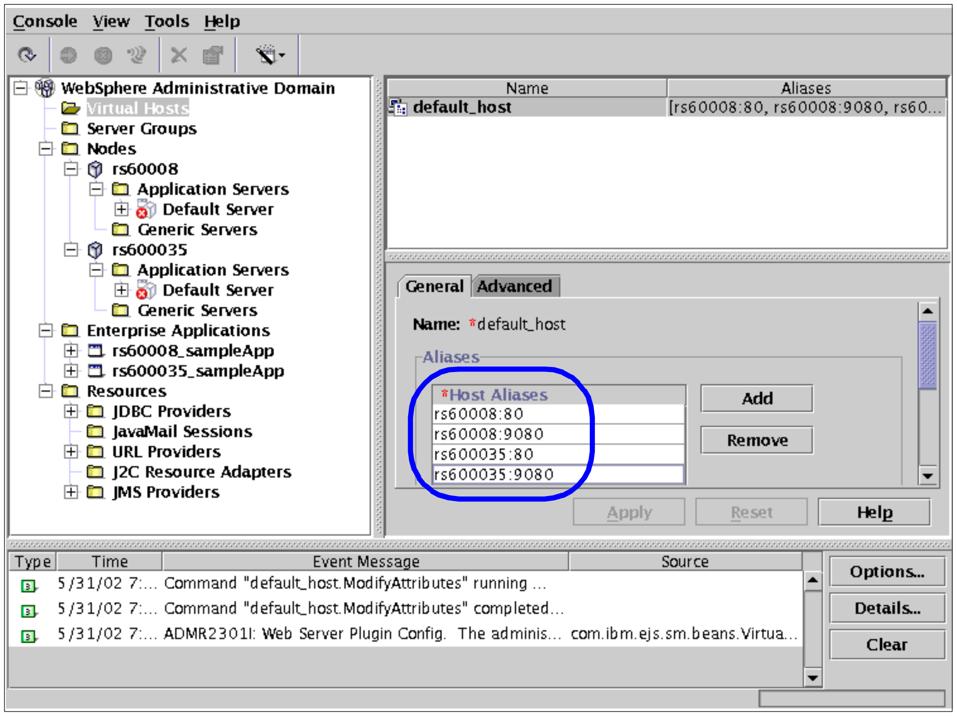
Aliases

*Host Aliases
*:80
*:9080

Figure 7-10   Default Host after WebSphere installation

You will see that the default host has two aliases configured. Both have a wildcard (asterisk) in the place of the hostname (see Figure 7-10). If we want to use multiple virtual hosts, we must change this to a defined hostname (usually the name of the system). Otherwise, the wildcard interferes with the other virtual hosts. If you do not intend to keep the Default Server, you may also delete it.

2. Change the asterisks to the hostname of your system (see Figure 7-11).



```
WebSphere Administrative Domain
  Virtual Hosts
  Server Groups
  Nodes
    rs60008
      Application Servers
        Default Server
      Generic Servers
    rs600035
      Application Servers
        Default Server
      Generic Servers
  Enterprise Applications
    rs60008_sampleApp
    rs600035_sampleApp
  Resources
    JDBC Providers
    JavaMail Sessions
    URL Providers
    J2C Resource Adapters
    JMS Providers
```

Name	Aliases
default_host	[rs60008:80, rs60008:9080, rs60...

Name: *default_host

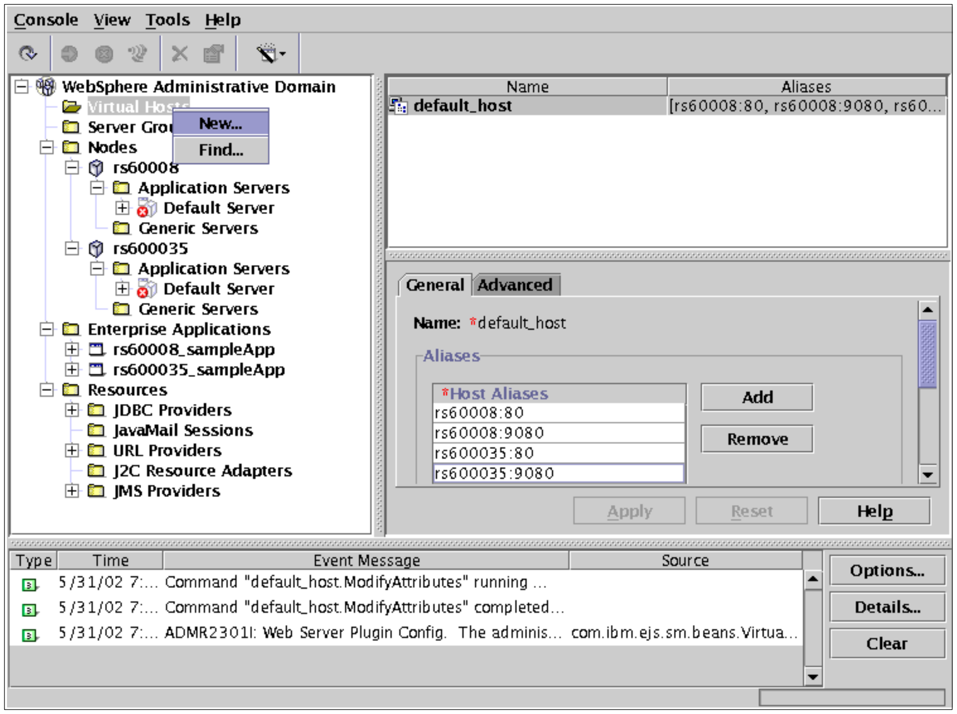
Aliases

*Host Aliases
rs60008:80
rs60008:9080
rs600035:80
rs600035:9080

```
Type  Time            Event Message                                      Source
      5/31/02 7:...   Command "default_host.ModifyAttributes" running ...
      5/31/02 7:...   Command "default_host.ModifyAttributes" completed...
      5/31/02 7:...   ADMR2301I: Web Server Plugin Config.  The adminis...  com.ibm.ejs.sm.beans.Virtua...
```

Figure 7-11 Added hostnames to the default host

3. Click **Apply** to save the changes.



Name	Aliases
default_host	[rs60008:80, rs60008:9080, rs60...

Name: *default_host

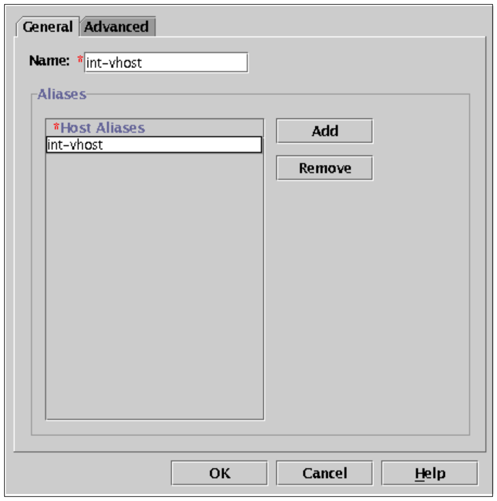
Aliases

*Host Aliases
rs60008:80
rs60008:9080
rs600035:80
rs600035:9080

```
Type  Time          Event Message                                      Source
      5/31/02 7:... Command "default_host.ModifyAttributes" running ...
      5/31/02 7:... Command "default_host.ModifyAttributes" completed...
      5/31/02 7:... ADMR2301I: Web Server Plugin Config.  The adminis... com.ibm.ejs.sm.beans.Virtua...
```

Figure 7-12 Create a new virtual host

4. Right-click the **Virtual Host** object in the left pane and select **New** (Virtual Host). Refer to Figure 7-12 and Figure 7-13.



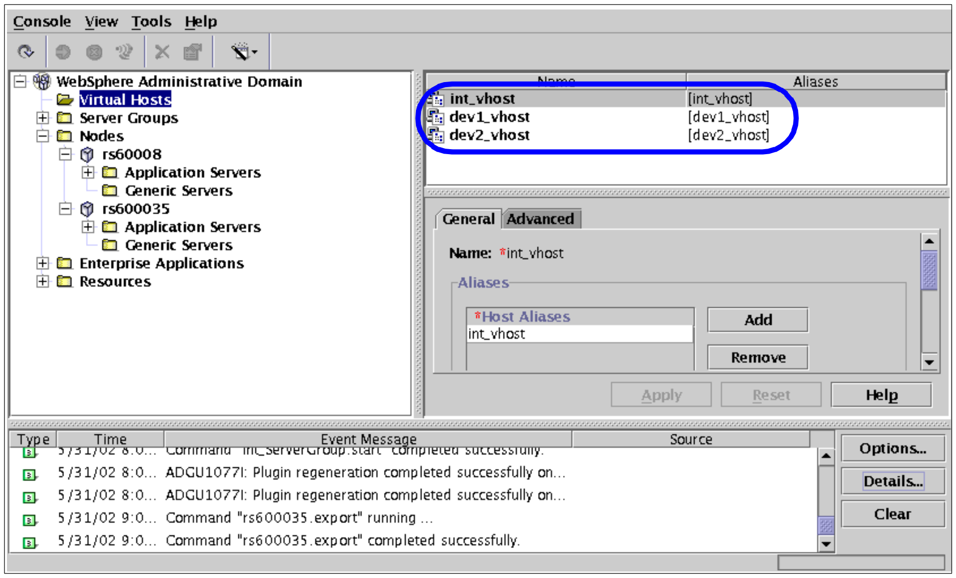
Name: *int-vhost

Aliases

*Host Aliases
int-vhost

Figure 7-13 Create new virtual host

5. Repeat these steps for each developer for which you want to create a testing space. In our scenario, there are three virtual hosts: (see Figure 7-13).
   - int_vhost
   - dev1_vhost
   - dev2_vhost



Name	Aliases
int_vhost	[int_vhost]
dev1_vhost	[dev1_vhost]
dev2_vhost	[dev2_vhost]

Name: *int_vhost

Aliases

*Host Aliases
int_vhost

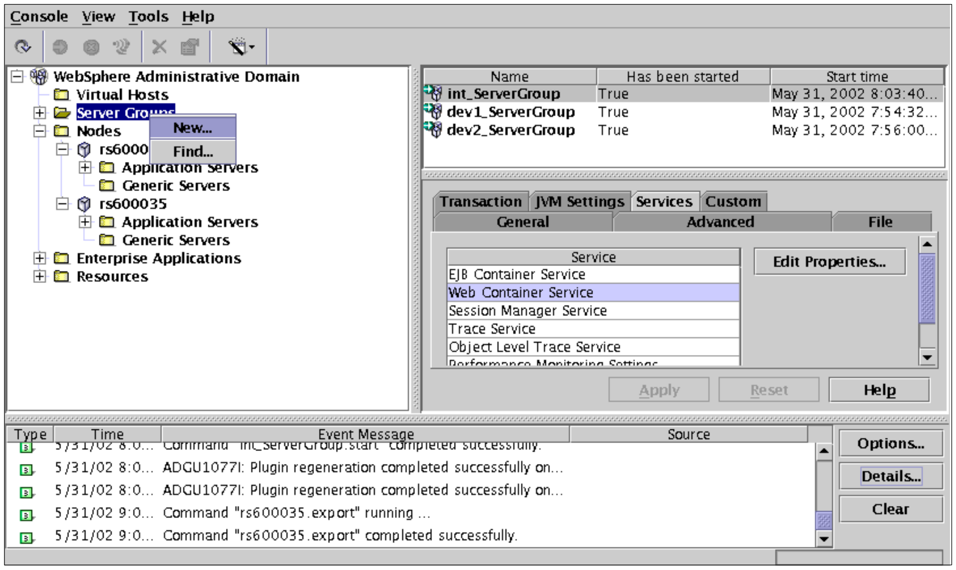
```
Type  Time          Event Message                                                     Source
      5/31/02 8:0...  Command "int_ServerGroup.start" completed successfully.
      5/31/02 8:0...  ADGU1077I: Plugin regeneration completed successfully on...
      5/31/02 8:0...  ADGU1077I: Plugin regeneration completed successfully on...
      5/31/02 9:0...  Command "rs600035.export" running ...
      5/31/02 9:0...  Command "rs600035.export" completed successfully.
```

Figure 7-14 Admin Console view after creation of three virtual hosts

Figure 7-14 displays the results of defining three virtual hosts.

Create ServerGroups

In this section, we create a ServerGroup for each workspace. Each ServerGroup will have two clones assigned to it: one on each Web Application Server node.



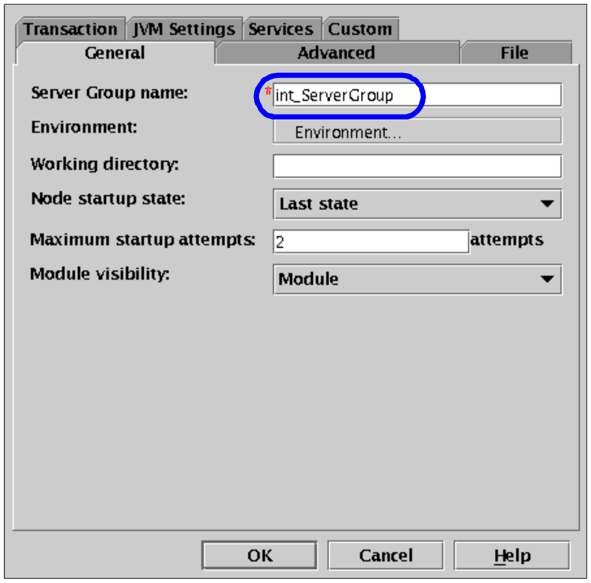
Name	Has been started	Start time
int_ServerGroup	True	May 31, 2002 8:03:40...
dev1_ServerGroup	True	May 31, 2002 7:54:32...
dev2_ServerGroup	True	May 31, 2002 7:56:00...

Service
EJB Container Service
Web Container Service
Session Manager Service
Trace Service
Object Level Trace Service
Performance Monitoring Settings

```
Type  Time         Event Message                                               Source
      5/31/02 8:0... Command "int_ServerGroup.start" completed successfully.
      5/31/02 8:0... ADGU1077I: Plugin regeneration completed successfully on...
      5/31/02 8:0... ADGU1077I: Plugin regeneration completed successfully on...
      5/31/02 9:0... Command "rs600035.export" running ...
      5/31/02 9:0... Command "rs600035.export" completed successfully.
```

Figure 7-15 Right-click ServerGroup

1. Right-click the **ServerGroup** object and select **New** (ServerGroup).

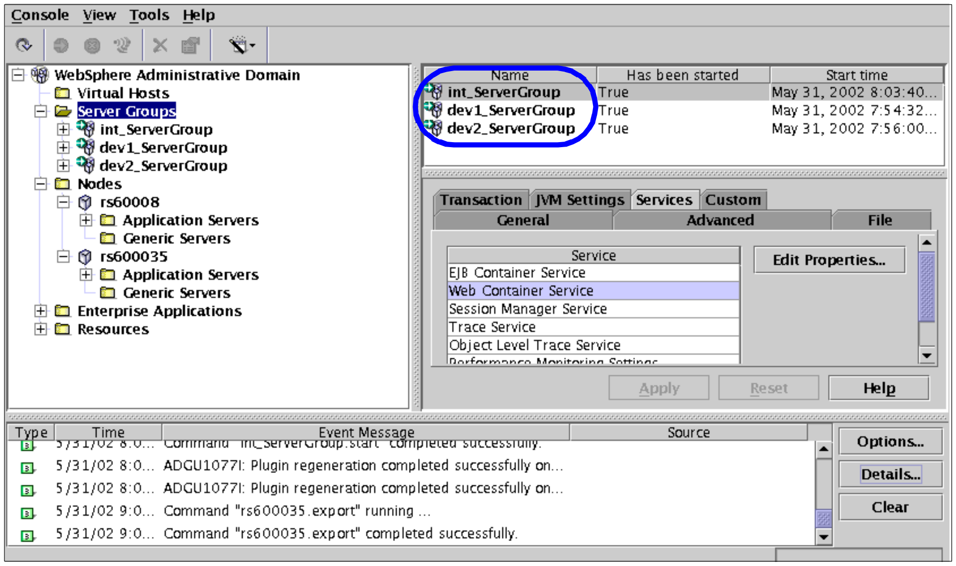


Server Group name:	int_ServerGroup
Environment:	Environment...
Working directory:	
Node startup state:	Last state
Maximum startup attempts:	2 attempts
Module visibility:	Module

Figure 7-16 Create new ServerGroup

2. Fill in the name of the ServerGroup. In our case, it is int_ServerGroup. Click **OK** to create the ServerGroup.

3. Repeat these steps for each developer. In our scenario, they are:
   - int_ServerGroup
   - dev1_ServerGroup
   - dev2_ServerGroup



Name	Has been started	Start time
int_ServerGroup	True	May 31, 2002 8:03:40...
dev1_ServerGroup	True	May 31, 2002 7:54:32...
dev2_ServerGroup	True	May 31, 2002 7:56:00...

Service
EJB Container Service
Web Container Service
Session Manager Service
Trace Service
Object Level Trace Service
Performance Monitoring Settings

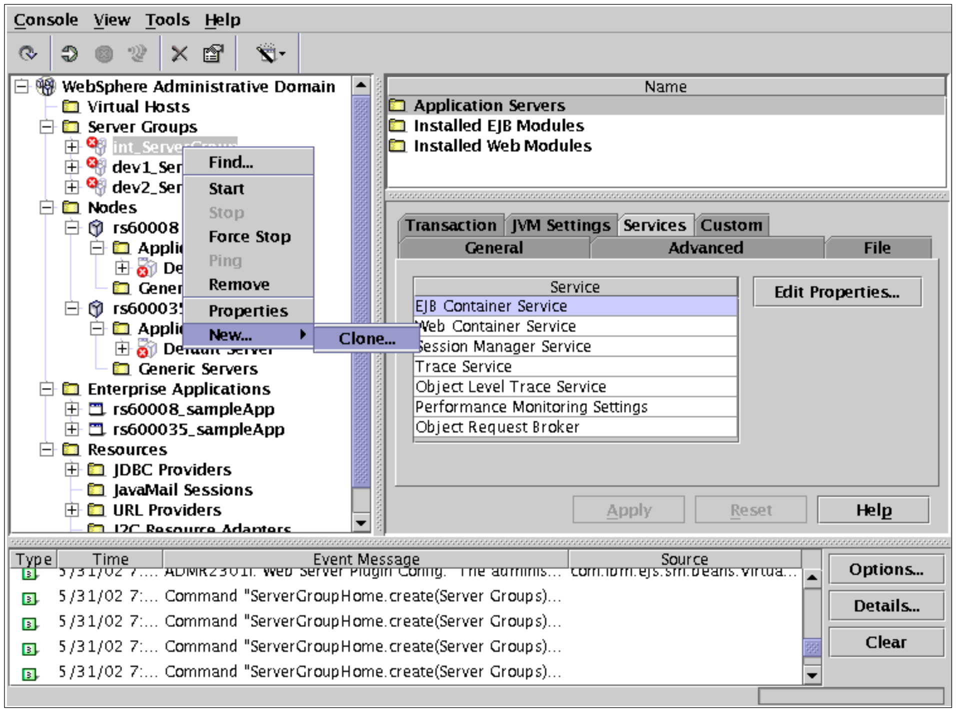
```
Type  Time          Event Message                                          Source
      5/31/02 8:0... Command "int_ServerGroup.start" completed successfully.
      5/31/02 8:0... ADGU1077I: Plugin regeneration completed successfully on...
      5/31/02 8:0... ADGU1077I: Plugin regeneration completed successfully on...
      5/31/02 9:0... Command "rs600035.export" running ...
      5/31/02 9:0... Command "rs600035.export" completed successfully.
```

Figure 7-17 The newly created ServerGroups

Figure 11-17 shows the ServerGroups we created.

Create clones to the ServerGroups

Now we will create two clones to each of the ServerGroups: one on each WebSphere Application Server node.

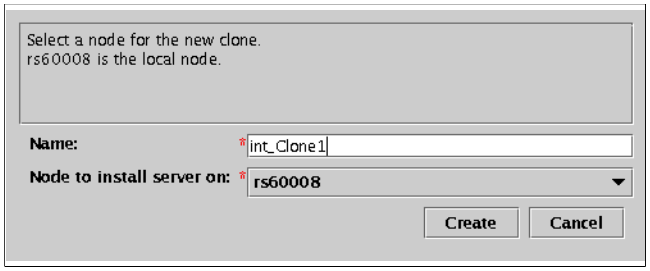


Service
EJB Container Service
Web Container Service
Session Manager Service
Trace Service
Object Level Trace Service
Performance Monitoring Settings
Object Request Broker

```
Type  Time        Event Message                                        Source
      5/31/02 7:... ADMR2301I: Web Server Plugin Config: The adminis... com.ibm.ejs.sm.beans.Virtua...
      5/31/02 7:... Command "ServerGroupHome.create(Server Groups)...
      5/31/02 7:... Command "ServerGroupHome.create(Server Groups)...
      5/31/02 7:... Command "ServerGroupHome.create(Server Groups)...
      5/31/02 7:... Command "ServerGroupHome.create(Server Groups)...
```

Figure 7-18 Right-click a ServerGroup

1. Right-click the ServerGroups you created and select **New -> Clone** (see Figure 7-18).


Select a node for the new clone.
rs60008 is the local node.

Name: * int_Clone1

Node to install server on: * rs60008

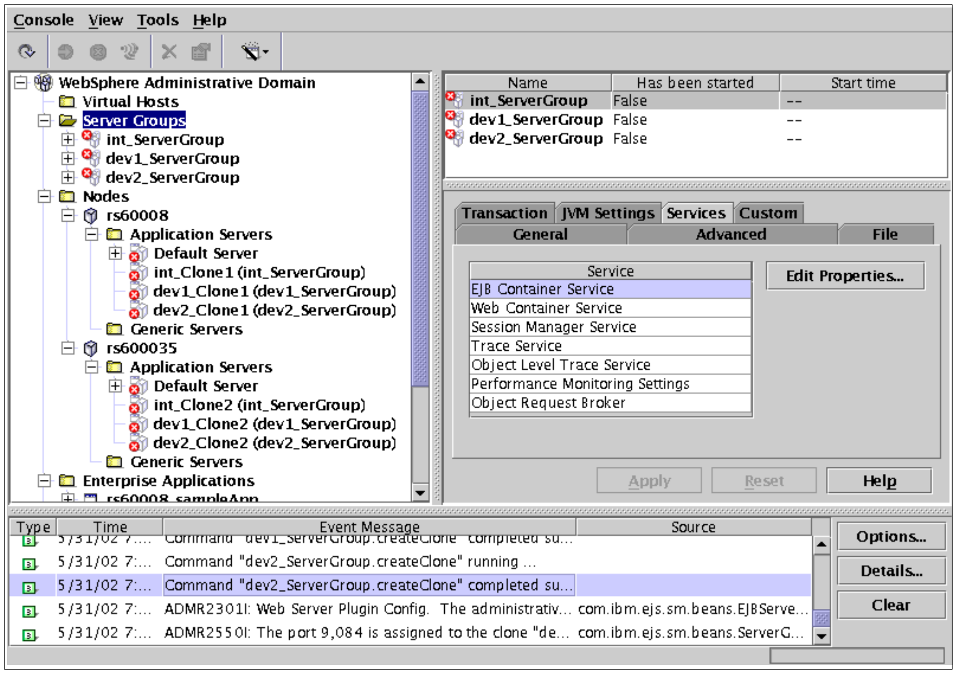
Figure 7-19 Naming of the new Clone

2. In the window that now appears, fill in the name of the new clone. Names of clones must be unique. Also, select on which node the clone will be placed. Then click **Create**.

3. Repeat these steps for each developer. In our scenario, they are:
   – int_Clone1, dev1_Clone1, dev2_Clone1 on node RS60008

   and

   – int_Clone2, dev1_Clone2, dev2_Clone2 on node RS600035



```
WebSphere Administrative Domain
    Virtual Hosts
    Server Groups
        int_ServerGroup
        dev1_ServerGroup
        dev2_ServerGroup
    Nodes
        rs60008
            Application Servers
                Default Server
                int_Clone1 (int_ServerGroup)
                dev1_Clone1 (dev1_ServerGroup)
                dev2_Clone1 (dev2_ServerGroup)
            Generic Servers
        rs600035
            Application Servers
                Default Server
                int_Clone2 (int_ServerGroup)
                dev1_Clone2 (dev1_ServerGroup)
                dev2_Clone2 (dev2_ServerGroup)
            Generic Servers
    Enterprise Applications
        rs60008_sampleApp
```

Name	Has been started	Start time
int_ServerGroup	False	--
dev1_ServerGroup	False	--
dev2_ServerGroup	False	--

Service
EJB Container Service
Web Container Service
Session Manager Service
Trace Service
Object Level Trace Service
Performance Monitoring Settings
Object Request Broker

Type	Time	Event Message	Source
	5/31/02 7:...	Command "dev1_ServerGroup.createClone" completed su...	
	5/31/02 7:...	Command "dev2_ServerGroup.createClone" running ...	
	5/31/02 7:...	Command "dev2_ServerGroup.createClone" completed su...	
	5/31/02 7:...	ADMR2301I: Web Server Plugin Config. The administrativ...	com.ibm.ejs.sm.beans.EJBServe...
	5/31/02 7:...	ADMR2550I: The port 9,084 is assigned to the clone "de...	com.ibm.ejs.sm.beans.ServerG...

Figure 7-20 The new clones

Figure 11-20 shows the ServerGroups and the clones.

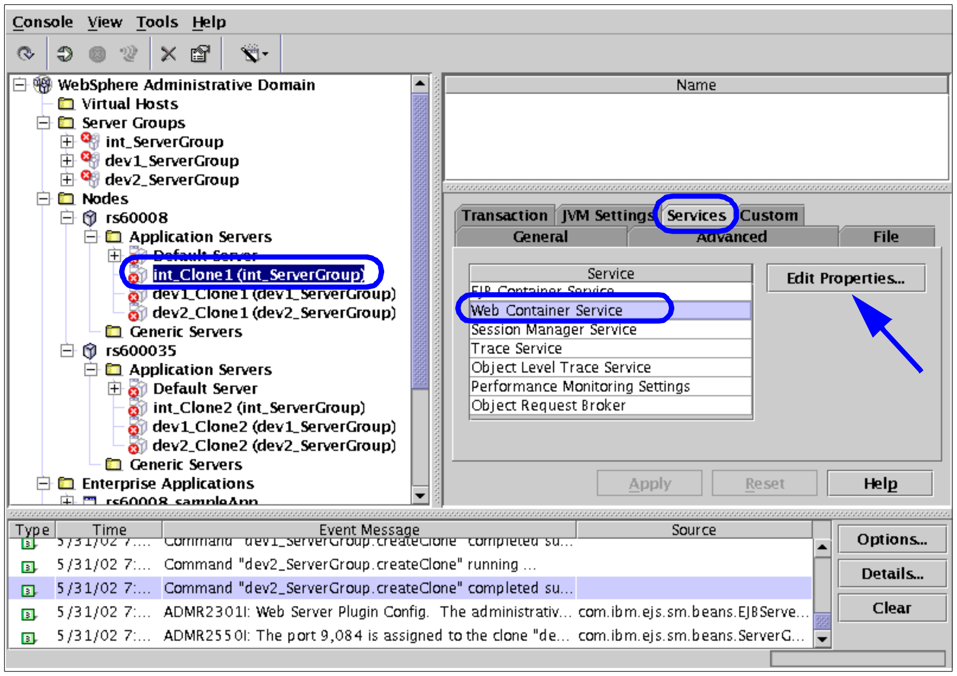
The clones we created in the last section have a wildcard hostname (asterisk) and a port number following a sequence assigned to them. For better organization and easier troubleshooting, we recommend that you apply some sort of policy to them. Table 7-3 shows the port numbers we used.

Table 7-3 Port numbers of the clones

Node	ServerGroup	Clone	Port
RS60008	int_ServerGroup	int_Clone1	8101
RS600035	int_ServerGroup	int_Clone2	8102
RS60008	dev1_ServerGroup	dev1_Clone1	8111

Node	ServerGroup	Clone	Port
RS600035	dev1_ServerGroup	dev1_Clone2	8112
RS60008	dev2_ServerGroup	dev2_Clone1	8121
RS600035	dev2_ServerGroup	dev2_Clone2	8122

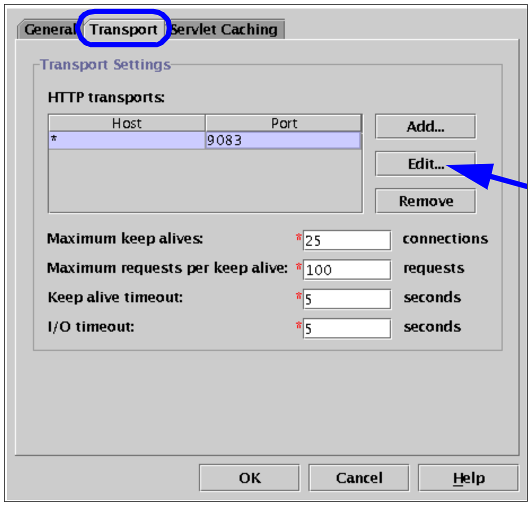
Transport configuration of the new clones



Type	Time	Event Message	Source
	5/31/02 7:...	Command "dev1_ServerGroup.createClone" completed su...	
	5/31/02 7:...	Command "dev2_ServerGroup.createClone" running ...	
	5/31/02 7:...	Command "dev2_ServerGroup.createClone" completed su...	
	5/31/02 7:...	ADMR2301I: Web Server Plugin Config. The administrativ...	com.ibm.ejs.sm.beans.EJBServe...
	5/31/02 7:...	ADMR2550I: The port 9,084 is assigned to the clone "de...	com.ibm.ejs.sm.beans.ServerG...

Figure 7-21 Select a clone

1. The next step is to configure a transport host and port for the clones. To do this, select a clone in the left-hand pane. Then, in the right-hand pane, select the **Services** tab. A list of services appears within this pane. Select **Web Container Service** and click **Edit Properties** as shown in Figure 7-21.



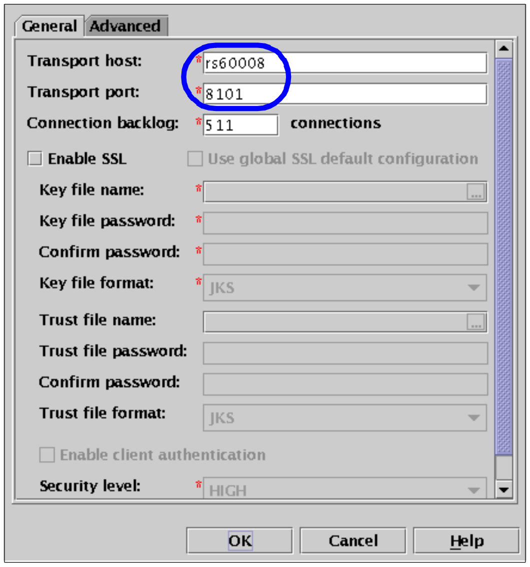
HTTP transports:

Host	Port
*	9083

Maximum keep alives:	25	connections
Maximum requests per keep alive:	100	requests
Keep alive timeout:	5	seconds
I/O timeout:	5	seconds

Figure 7-22 Web container properties

2. The properties window appears. First select the **Transport** tab. Then select the **HTTP transport** entry and click **Edit.**

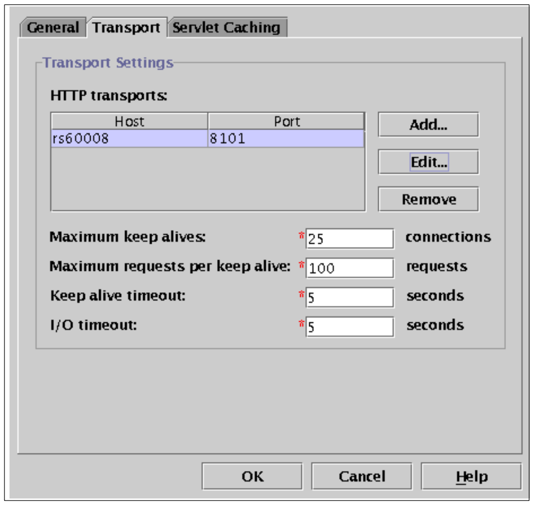


Transport host:	rs60008	
Transport port:	8101	
Connection backlog:	511	connections
☐ Enable SSL	☐ Use global SSL default configuration	
Key file name:		
Key file password:		
Confirm password:		
Key file format:	JKS	
Trust file name:		
Trust file password:		
Confirm password:		
Trust file format:	JKS	
☐ Enable client authentication		
Security level:	HIGH	

Figure 7-23 Web Container Transport properties

3. Fill in the the **Transport host** field. This is the node name on which the clone resides.

4. Check that the chosen **Transport port** is available. You are free to choose any free port number.
5. Click **OK** to save your settings (see Figure 7-23).



Transport Settings

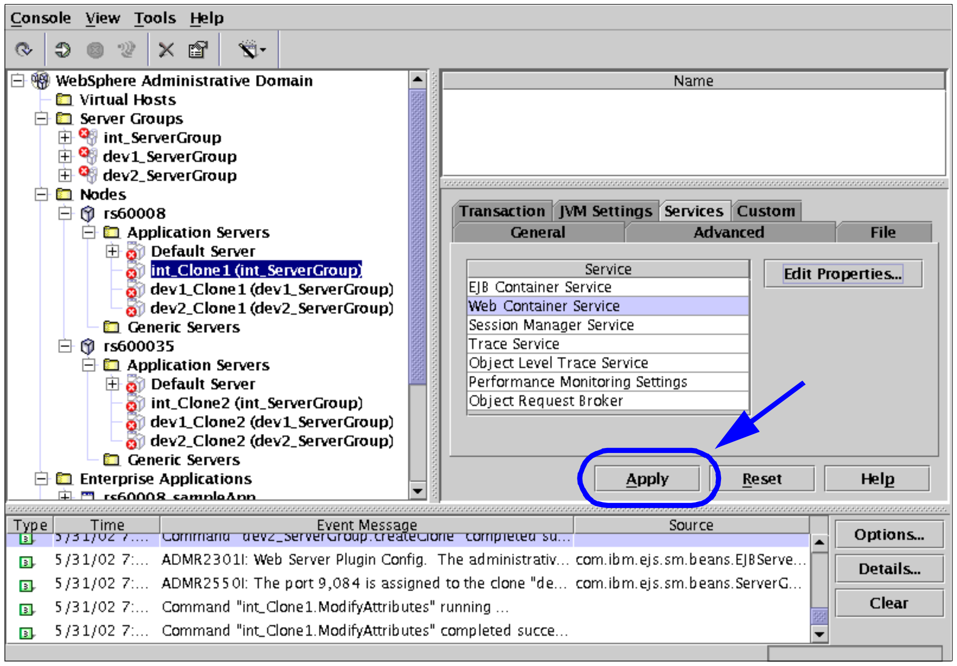
HTTP transports:

Host	Port
rs60008	8101

Maximum keep alives: * 25 connections
Maximum requests per keep alive: * 100 requests
Keep alive timeout: * 5 seconds
I/O timeout: * 5 seconds

Figure 7-24 Back to the clone properties window

6. The clones properties window now shows the host and port you specified. Click **OK** to save the changes.



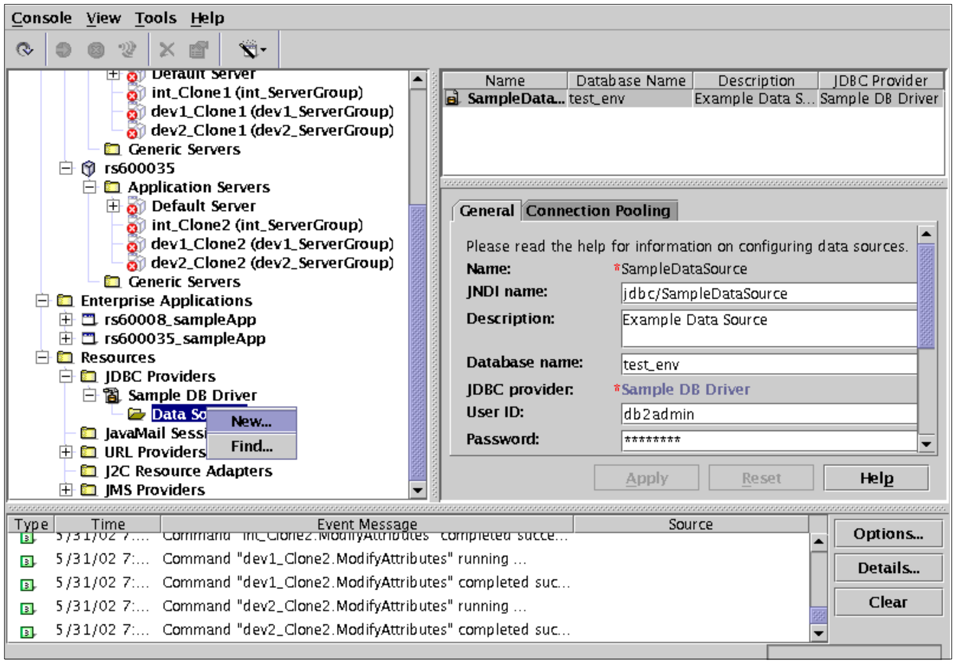
Type	Time	Event Message	Source
	5/31/02 7:...	Command "dev2_ServerGroup.createClone" completed su...	
	5/31/02 7:...	ADMR2301I: Web Server Plugin Config. The administrativ...	com.ibm.ejs.sm.beans.EJBServe...
	5/31/02 7:...	ADMR2550I: The port 9,084 is assigned to the clone "de...	com.ibm.ejs.sm.beans.ServerG...
	5/31/02 7:...	Command "int_Clone1.ModifyAttributes" running ...	
	5/31/02 7:...	Command "int_Clone1.ModifyAttributes" completed succe...	

Figure 7-25 Back to the main window

7. Back in the main window, make sure you click **Apply** to save the changes.

Create new Data Sources

To allow the developers to test their applications without interfering with others, we created a database for each of them. To access them, we need Data Sources. Each Data Source will point to a separate database.



Name	Database Name	Description	JDBC Provider
SampleData...	test_env	Example Data S...	Sample DB Driver

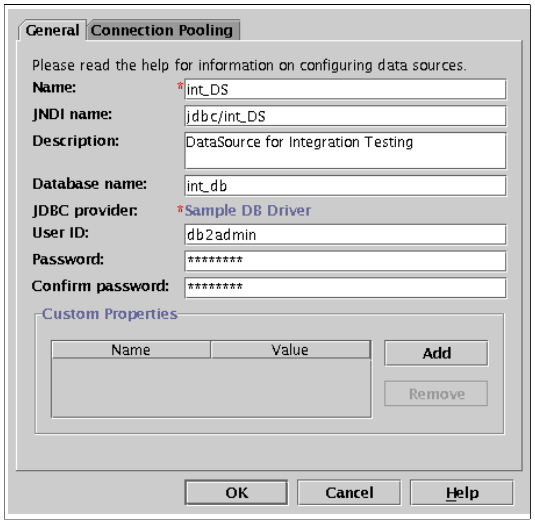
Please read the help for information on configuring data sources.

Name:	*SampleDataSource
JNDI name:	jdbc/SampleDataSource
Description:	Example Data Source
Database name:	test_env
JDBC provider:	*Sample DB Driver
User ID:	db2admin
Password:	********

```
Type  Time          Event Message                                              Source
      5/31/02 7:... Command "int_Clone2.ModifyAttributes" completed succe...
      5/31/02 7:... Command "dev1_Clone2.ModifyAttributes" running ...
      5/31/02 7:... Command "dev1_Clone2.ModifyAttributes" completed suc...
      5/31/02 7:... Command "dev2_Clone2.ModifyAttributes" running ...
      5/31/02 7:... Command "dev2_Clone2.ModifyAttributes" completed suc...
```

Figure 7-26 Open the Resources tree in the left-hand pane

1. In the Administration Console, go to the left-hand pane and click **Resources -> JDBC Providers -> Sample DB Driver -> Data Sources**. Right-click **Data Sources** and select **New** from the menu. See Figure 7-26.


General | Connection Pooling

Please read the help for information on configuring data sources.

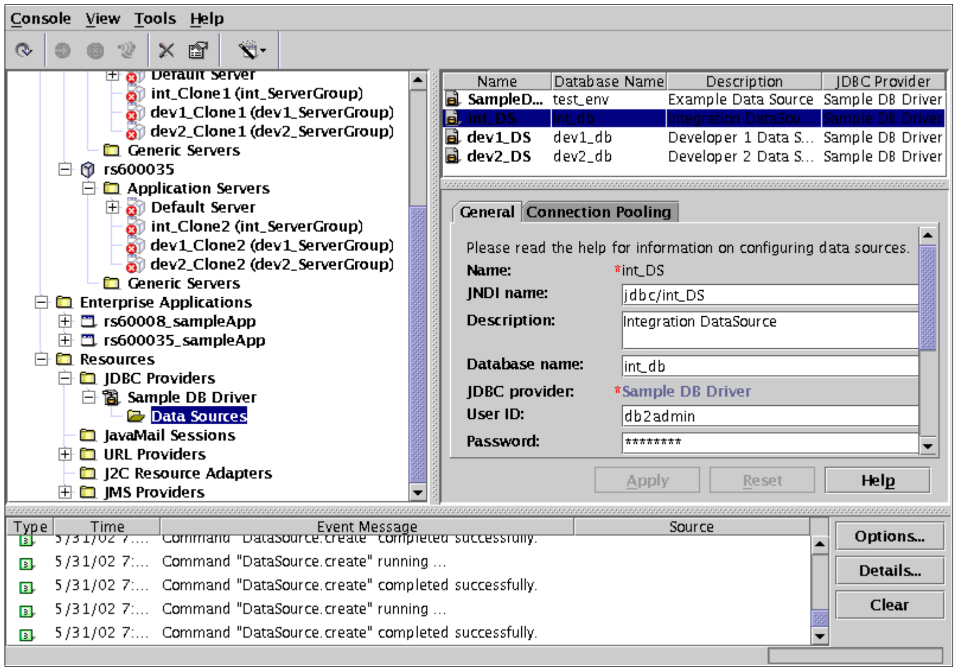
Name:	*int_DS
JNDI name:	jdbc/int_DS
Description:	DataSource for Integration Testing
Database name:	int_db
JDBC provider:	*Sample DB Driver
User ID:	db2admin
Password:	********
Confirm password:	********

Custom Properties

Name	Value

Figure 7-27 Create Data Source

2. Fill in the name, Java Naming and Directory Interface (JNDI) name, description, DB name and DB user. In our example, these are:
   – Name: `int_DS`
   – JNDI name: `jdbc/int_DS`
   – Description
   – User ID: `db2admin`
   – Password: db2admin password
3. Click **OK** to create the Data Source.
4. Repeat the steps for each developer. In our example, they are:
   – dev1_DS, pointing to dev1_db, jdbc/dev1_DS, db2admin
   – dev2_DS, pointing to dev2_db, jdbc/dev2_DS, db2admin



- Default Server
- int_Clone1 (int_ServerGroup)
- dev1_Clone1 (dev1_ServerGroup)
- dev2_Clone1 (dev2_ServerGroup)
- Generic Servers
- rs600035
  - Application Servers
    - Default Server
    - int_Clone2 (int_ServerGroup)
    - dev1_Clone2 (dev1_ServerGroup)
    - dev2_Clone2 (dev2_ServerGroup)
  - Generic Servers
- Enterprise Applications
  - rs60008_sampleApp
  - rs600035_sampleApp
- Resources
  - JDBC Providers
    - Sample DB Driver
      - Data Sources
  - JavaMail Sessions
  - URL Providers
  - J2C Resource Adapters
  - JMS Providers

Name	Database Name	Description	JDBC Provider
SampleD...	test_env	Example Data Source	Sample DB Driver
int_DS	int_db	Integration DataSou...	Sample DB Driver
dev1_DS	dev1_db	Developer 1 Data S...	Sample DB Driver
dev2_DS	dev2_db	Developer 2 Data S...	Sample DB Driver

General | Connection Pooling

Please read the help for information on configuring data sources.

Name:	*int_DS
JNDI name:	jdbc/int_DS
Description:	Integration DataSource
Database name:	int_db
JDBC provider:	*Sample DB Driver
User ID:	db2admin
Password:	********

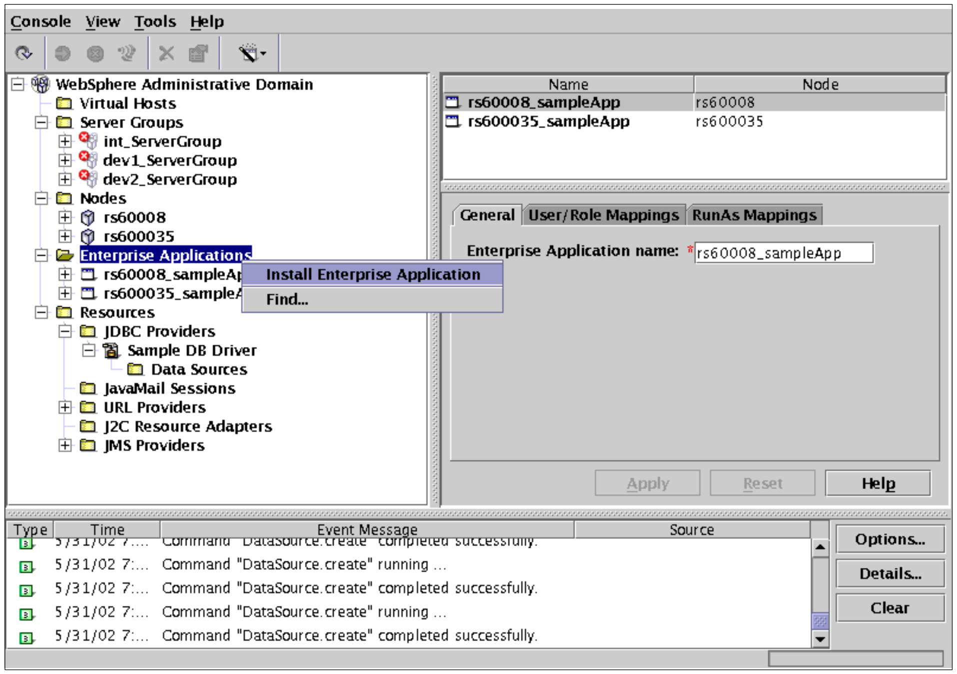
Type	Time	Event Message	Source
	5/31/02 7:...	Command "DataSource.create" completed successfully.	
	5/31/02 7:...	Command "DataSource.create" running ...	
	5/31/02 7:...	Command "DataSource.create" completed successfully.	
	5/31/02 7:...	Command "DataSource.create" running ...	
	5/31/02 7:...	Command "DataSource.create" completed successfully.	

Figure 7-28 The new Data Sources

Figure 7-28 shows the Data Sources after completion.

Installation of the same application for each developer

To test the created environments, we now install the same application for each developer's ServerGroup. We choose the sampleApp application that is provided with each WebSphere installation.



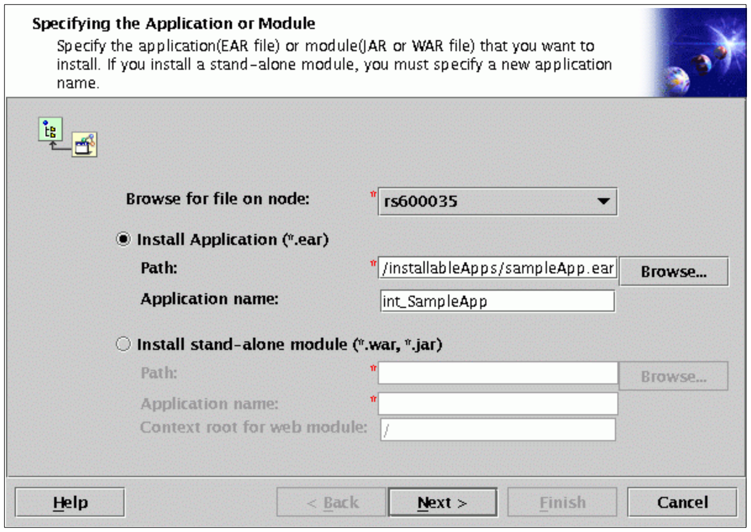
Name	Node
rs60008_sampleApp	rs60008
rs600035_sampleApp	rs600035

Enterprise Application name: rs60008_sampleApp

Type	Time	Event Message	Source
	5/31/02 7:...	Command "DataSource.create" completed successfully.	
	5/31/02 7:...	Command "DataSource.create" running ...	
	5/31/02 7:...	Command "DataSource.create" completed successfully.	
	5/31/02 7:...	Command "DataSource.create" running ...	
	5/31/02 7:...	Command "DataSource.create" completed successfully.	

Figure 7-29 Installing an Enterprise Application

1. In the left-hand pane, right-click **Enterprise Applications** and select **Install Enterprise Application**.


Specifying the Application or Module

Specify the application(EAR file) or module(JAR or WAR file) that you want to install. If you install a stand-alone module, you must specify a new application name.

Browse for file on node: rs600035

[✓ Install Application (*.ear)]

Path: /installableApps/sampleApp.ear

Application name: int_SampleApp

Install stand-alone module (*.war, *.jar)

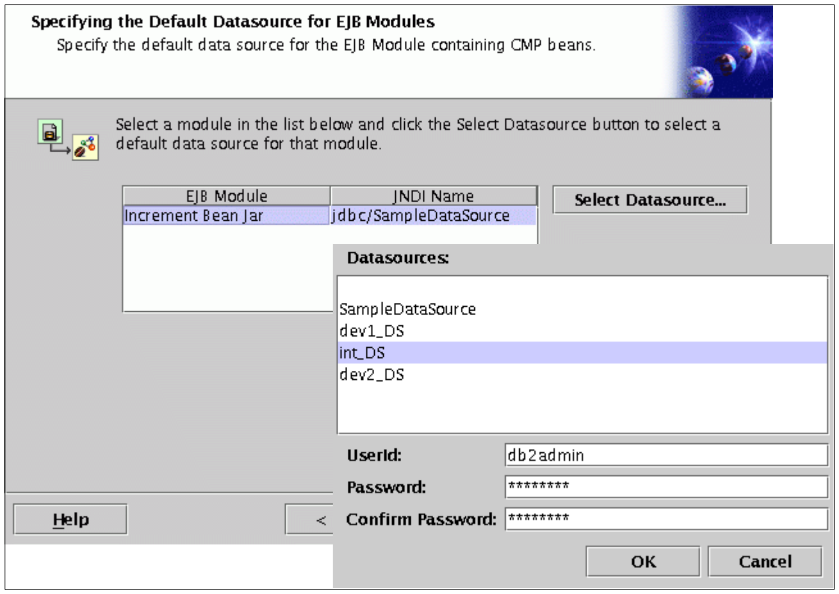
Path:

Application name:

Context root for web module: /

Figure 7-30 Specify the application to be installed

2. Browse to the directory where the application *.ear file is located and select it. We chose the sampleApp application which is located in the WebSphere installableApps directory. Specify a display name for it. This name has no influence on the application's runtime behavior but has to be unique within the WebSphere domain. It is only used in the Administrative console and for the directory which will be created in the WebSphere installedApps directory. We chose the name int_SampleApp (see Figure 7-30).
3. Click **Next** through the next few windows. These allow you to view and alter user mappings, security settings, JNDI mappings and so on. These settings are normally dealt with by the developer that packages the application. So, in most cases, you can click through them.



Specifying the Default Datasource for EJB Modules

Specify the default data source for the EJB Module containing CMP beans.

Select a module in the list below and click the Select Datasource button to select a default data source for that module.

EJB Module	JNDI Name
Increment Bean Jar	jdbc/SampleDataSource

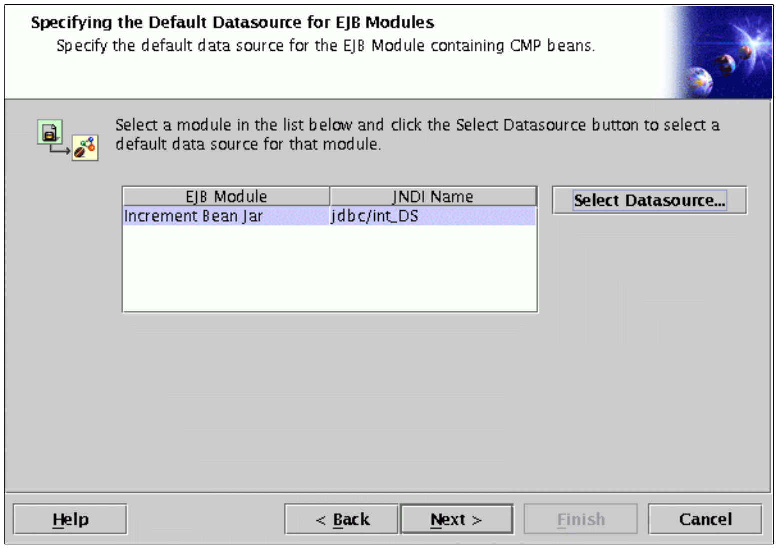
Datasources:

SampleDataSource
dev1_DS
int_DS
dev2_DS

UserId: db2admin
Password: ********
Confirm Password: ********

Figure 7-31 Select default Data Source for the CMP EJB module

4. When you arrive at the Specify Default DataSource window, click **Select DataSource** to change the Data Source to the one you created for this developer. Fill in database user ID and password. Click **OK** to save the changes. Refer to Figure 7-31.


Specifying the Default Datasource for EJB Modules

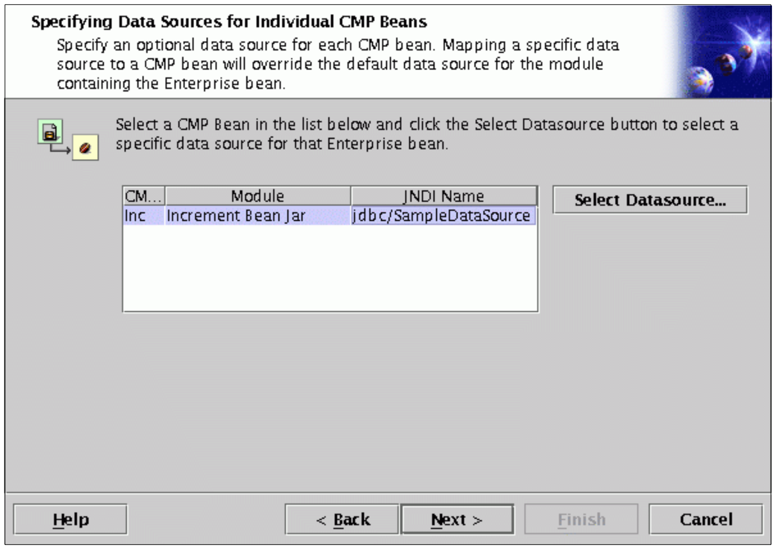
Specify the default data source for the EJB Module containing CMP beans.

Select a module in the list below and click the Select Datasource button to select a default data source for that module.

EJB Module	JNDI Name
Increment Bean Jar	jdbc/int_DS

Figure 7-32 The changed default Data Source

5. Figure 7-32 shows the JNDI name of the selected Data Source.


Specifying Data Sources for Individual CMP Beans

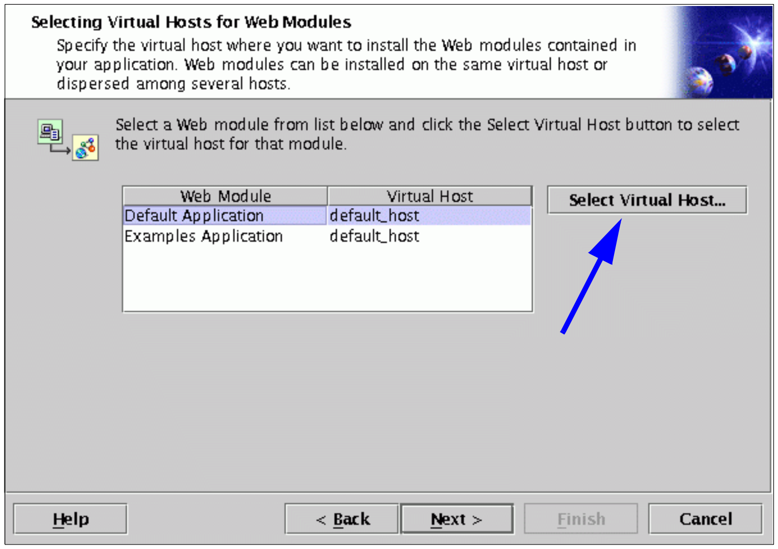
Specify an optional data source for each CMP bean. Mapping a specific data source to a CMP bean will override the default data source for the module containing the Enterprise bean.

Select a CMP Bean in the list below and click the Select Datasource button to select a specific data source for that Enterprise bean.

CM...	Module	JNDI Name
Inc	Increment Bean Jar	jdbc/SampleDataSource

Figure 7-33 Select Data Source for individual CMB Beans

6. The next window (Figure 7-33) shows all CMP Beans. You can select a distinct Data Source for each CMB Bean. Click **Select Datasource** and select the datasource we created for this developer.



Selecting Virtual Hosts for Web Modules

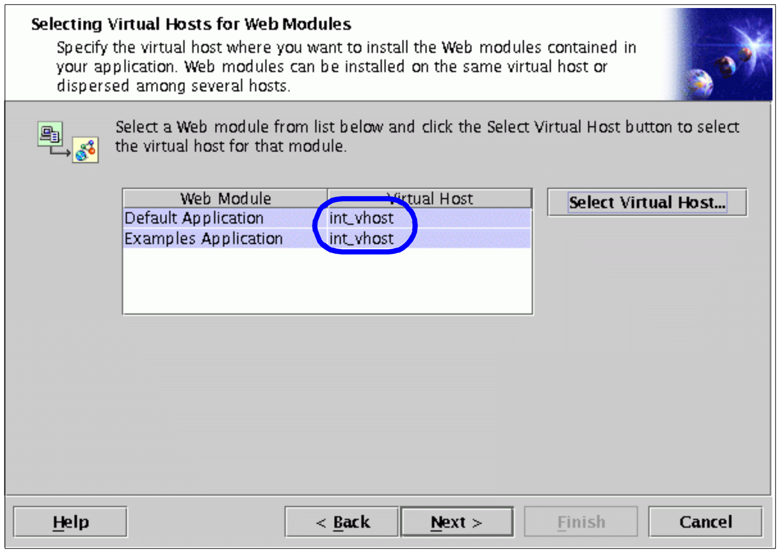
Specify the virtual host where you want to install the Web modules contained in your application. Web modules can be installed on the same virtual host or dispersed among several hosts.

Select a Web module from list below and click the Select Virtual Host button to select the virtual host for that module.

Web Module	Virtual Host
Default Application	default_host
Examples Application	default_host

Figure 7-34   Selecting a virtual host for Web Modules

7. In the window shown in Figure 7-34, you can select a virtual host for each Web module of the application. Click **Select Virtual Host** to change the default setting.



Selecting Virtual Hosts for Web Modules

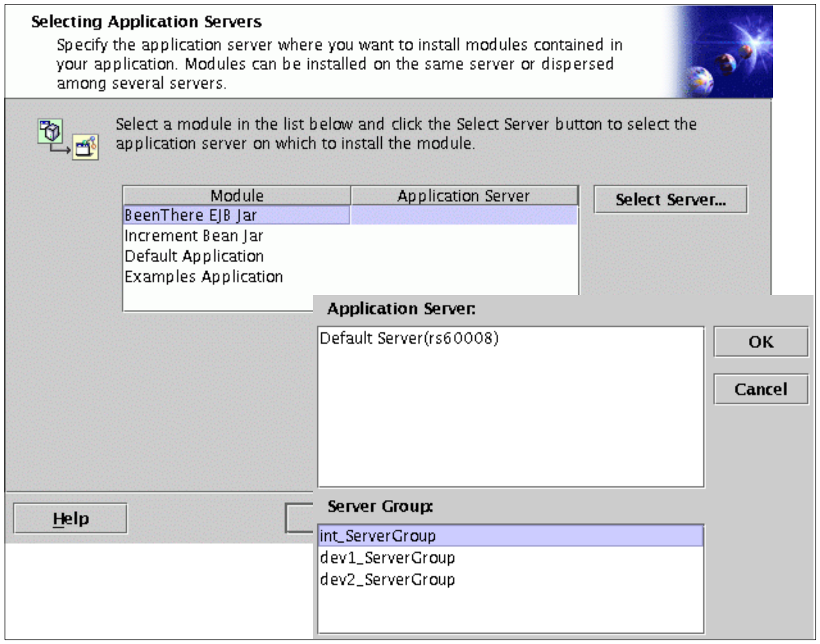
Specify the virtual host where you want to install the Web modules contained in your application. Web modules can be installed on the same virtual host or dispersed among several hosts.

Select a Web module from list below and click the Select Virtual Host button to select the virtual host for that module.

Web Module	Virtual Host
Default Application	int_vhost
Examples Application	int_vhost

Figure 7-35   Changed Virtual Hosts for Web modules

8. We changed all Web modules to the int_vhost virtual host. Click **Next** to proceed.


Selecting Application Servers

Specify the application server where you want to install modules contained in your application. Modules can be installed on the same server or dispersed among several servers.

Select a module in the list below and click the Select Server button to select the application server on which to install the module.

Module	Application Server
BeenThere EJB Jar	
Increment Bean Jar	
Default Application	
Examples Application	

Application Server:

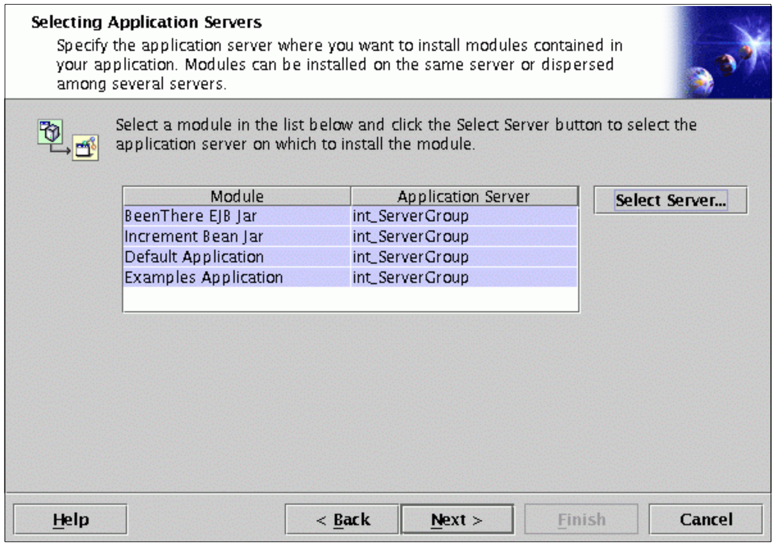
Default Server(rs60008)

Server Group:

int_ServerGroup
dev1_ServerGroup
dev2_ServerGroup

Figure 7-36 Selecting Application Servers for all modules

9. In this window, an Application Server or a ServerGroup must be selected for each module. We selected the int_ServerGroup for all modules.


Selecting Application Servers

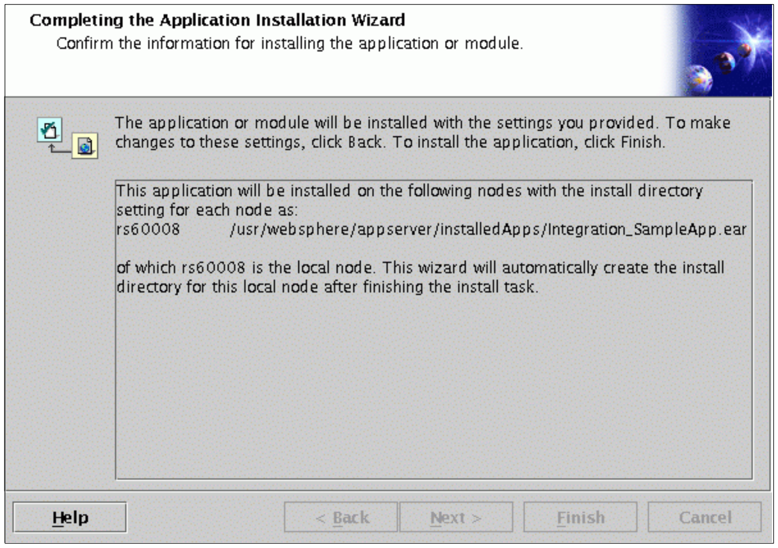
Specify the application server where you want to install modules contained in your application. Modules can be installed on the same server or dispersed among several servers.

Select a module in the list below and click the Select Server button to select the application server on which to install the module.

Module	Application Server
BeenThere EJB Jar	int_ServerGroup
Increment Bean Jar	int_ServerGroup
Default Application	int_ServerGroup
Examples Application	int_ServerGroup

Figure 7-37 The selected Application Servers

10. Figure 7-37 shows the selected Application Servers for each module. Click **Next** to proceed.


Completing the Application Installation Wizard

Confirm the information for installing the application or module.

The application or module will be installed with the settings you provided. To make changes to these settings, click Back. To install the application, click Finish.

```
This application will be installed on the following nodes with the install directory
setting for each node as:
rs60008       /usr/websphere/appserver/installedApps/Integration_SampleApp.ear

of which rs60008 is the local node. This wizard will automatically create the install
directory for this local node after finishing the install task.
```

Figure 7-38 Possible regeneration of code

11. Now, all necessary information is gathered to install the application. Click **Finish** to proceed.

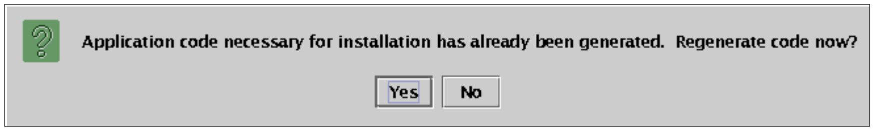

Application code necessary for installation has already been generated. Regenerate code now?

Figure 7-39 Code regeneration

12. You will be asked wether you want to regenerate your code. If you do so, additional panels, not shown here, will appear.

13. The application is now being installed and deployed. The *.ear file will be expanded to the WebSphere installedApps directory on the node you chose from the original *.ear archive. If you plan to run this application on multiple nodes, you will need to distribute the created directory to all the other nodes.

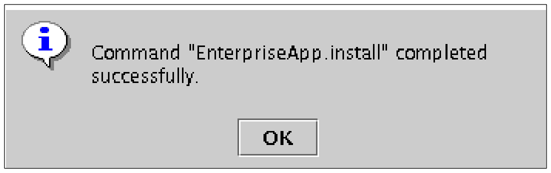
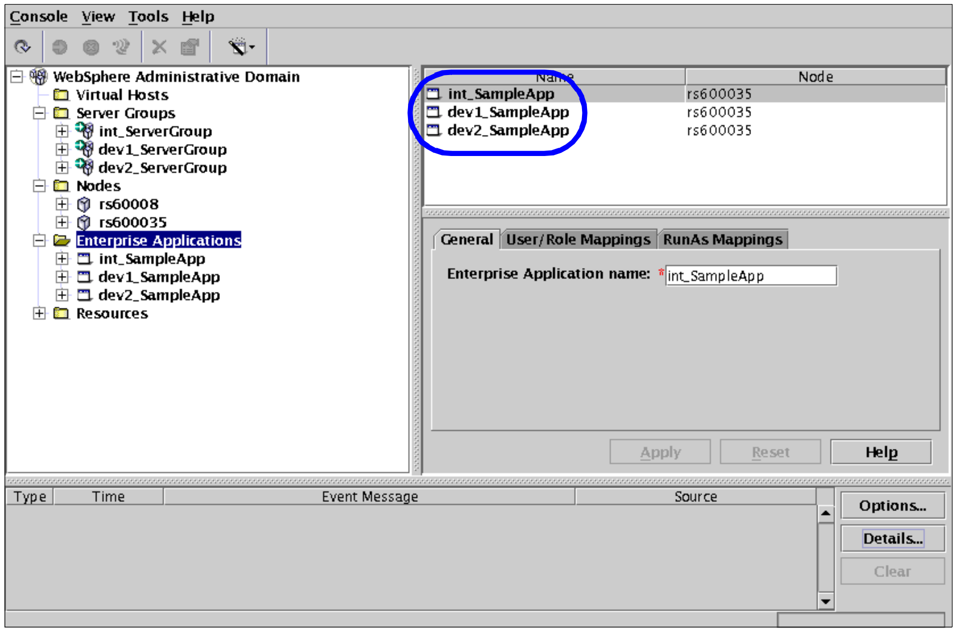

Command "EnterpriseApp.install" completed successfully.

Figure 7-40 Completion of application installation

14. If the installation was successful, the completion window appears. Click **OK** to confirm.

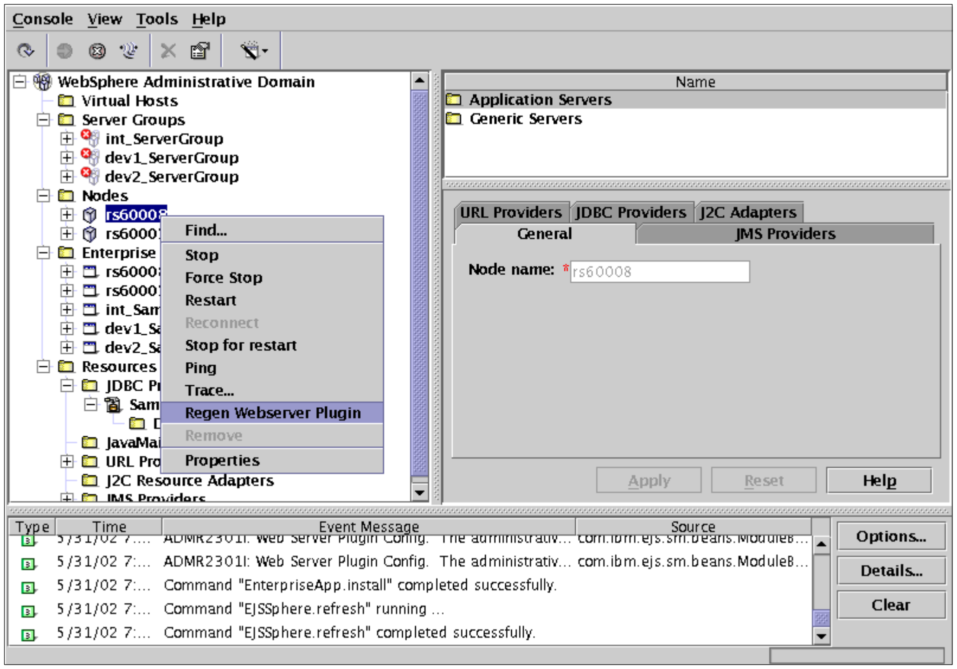


Name	Node
int_SampleApp	rs600035
dev1_SampleApp	rs600035
dev2_SampleApp	rs600035

Enterprise Application name: * int_SampleApp

Figure 7-41 The same application installed into three distinct environments

15. Repeat these steps for each developer environment. In our example, the applications are:
    - IntegrationSampleApp
    - dev1_SampleApp
    - dev2_SampleApp

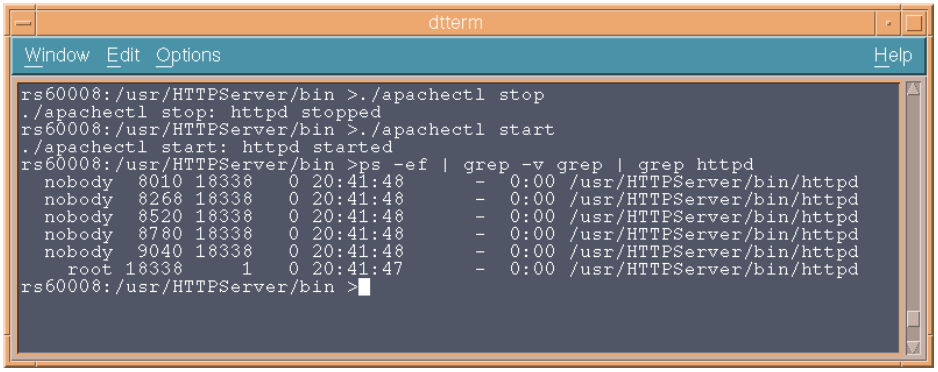


Node name: * rs60008

```
Type  Time          Event Message                                              Source
      5/31/02 7:... ADMR2301I: Web Server Plugin Config.  The administrativ... com.ibm.ejs.sm.beans.ModuleB...
      5/31/02 7:... ADMR2301I: Web Server Plugin Config.  The administrativ... com.ibm.ejs.sm.beans.ModuleB...
      5/31/02 7:... Command "EnterpriseApp.install" completed successfully.
      5/31/02 7:... Command "EJSSphere.refresh" running ...
      5/31/02 7:... Command "EJSSphere.refresh" completed successfully.
```

Figure 7-42 Regenerate plug-in config files

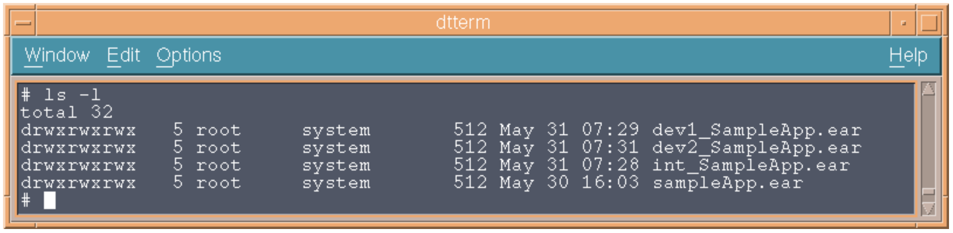
16. Each time you install or uninstall an application, the plug-in information must be updated. Regenerate the plug-in configuration files for the WebServer plug-in on each node. Right-click the node and select **Regen Webserver Plugin**.



```
rs60008:/usr/HTTPServer/bin >./apachectl stop
./apachectl stop: httpd stopped
rs60008:/usr/HTTPServer/bin >./apachectl start
./apachectl start: httpd started
rs60008:/usr/HTTPServer/bin >ps -ef | grep -v grep | grep httpd
  nobody  8010 18338   0 20:41:48      -  0:00 /usr/HTTPServer/bin/httpd
  nobody  8268 18338   0 20:41:48      -  0:00 /usr/HTTPServer/bin/httpd
  nobody  8520 18338   0 20:41:48      -  0:00 /usr/HTTPServer/bin/httpd
  nobody  8780 18338   0 20:41:48      -  0:00 /usr/HTTPServer/bin/httpd
  nobody  9040 18338   0 20:41:48      -  0:00 /usr/HTTPServer/bin/httpd
    root 18338     1   0 20:41:47      -  0:00 /usr/HTTPServer/bin/httpd
rs60008:/usr/HTTPServer/bin >
```

Figure 7-43 Restart of IBM HTTP Server

17. After the plug-in config regeneration, you need to restart the Web Server on each node.



```
# ls -l
total 32
drwxrwxrwx   5 root     system       512 May 31 07:29 dev1_SampleApp.ear
drwxrwxrwx   5 root     system       512 May 31 07:31 dev2_SampleApp.ear
drwxrwxrwx   5 root     system       512 May 31 07:28 int_SampleApp.ear
drwxrwxrwx   5 root     system       512 May 30 16:03 sampleApp.ear
#
```

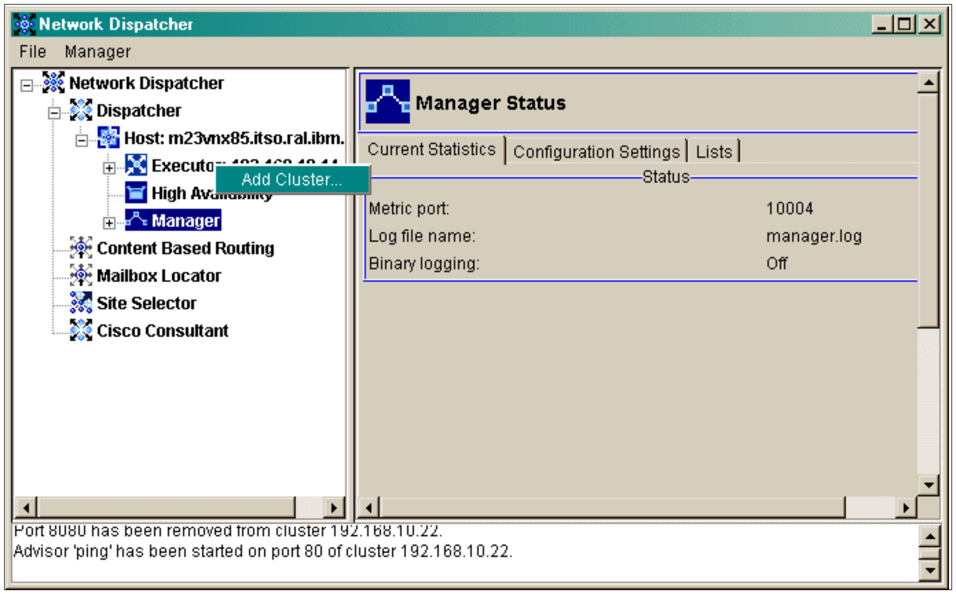
Figure 7-44 installedApps directory

18. Figure 7-44 shows the WebSphere installedApps directory after the installation of all three applications.
19. The last step is to distribute these directories to all Websphere nodes that will serve this application. The easiest way is to create a *.tar archive, FTP it to the other nodes and un-tar it there.

7.2.3 Load balancer node

The last step is to configure the load balancer node to balance the requests between the members of the cluster. The cluster IP address is 192.168.10.50.

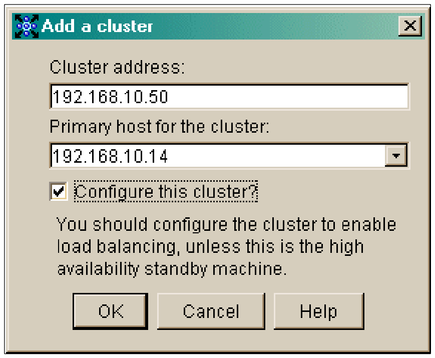
1. Start the Network Dispatcher GUI and connect to the host by right-clicking the **Dispatcher** item and selecting **Connect to Host**. Once you are connected, start the Executor. The Executor will show the server's IP address when it has started. Now right-click the **Executor** item and select **Add Cluster**. See Figure 7-45.



Metric port: 10004
Log file name: manager.log
Binary logging: Off

```
Port 8080 has been removed from cluster 192.168.10.22.
Advisor 'ping' has been started on port 80 of cluster 192.168.10.22.
```

Figure 7-45 Add a cluster



Cluster address: 192.168.10.50

Primary host for the cluster: 192.168.10.14

☑ Configure this cluster?

You should configure the cluster to enable load balancing, unless this is the high availability standby machine.

Figure 7-46 Specify the cluster address

2. In the Add a cluster window, specify the IP address of the cluster. Leave the primary host as it is suggested. Select the **Configure this cluster?** box and click **OK**.

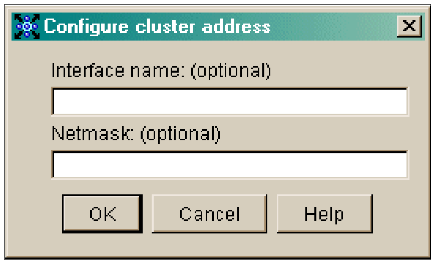
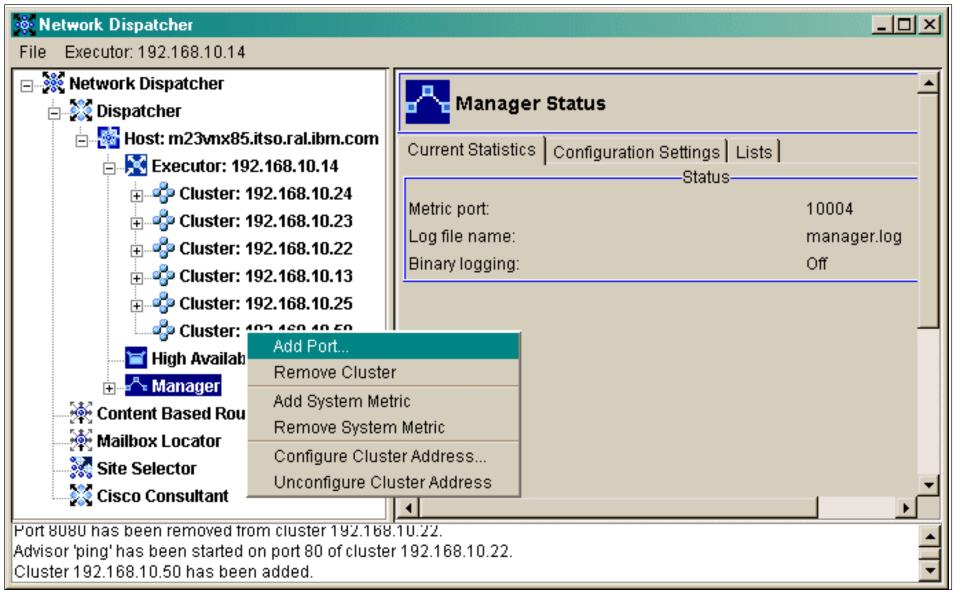


Figure 7-47 Configure the cluster

3. If you leave the Interface and Netmask fields blank, the first interface will be chosen.



Metric port:	10004
Log file name:	manager.log
Binary logging:	Off

```
Port 8080 has been removed from cluster 192.168.10.22.
Advisor 'ping' has been started on port 80 of cluster 192.168.10.22.
Cluster 192.168.10.50 has been added.
```

Figure 7-48 Add a port to the cluster

4. The next step is to add a port number to the cluster. Right-click the cluster item you created and select **Add Port**.

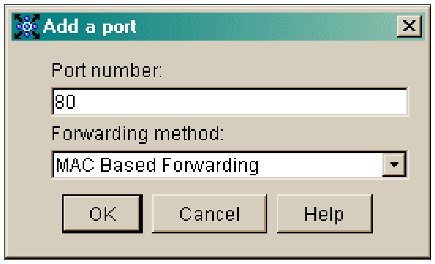
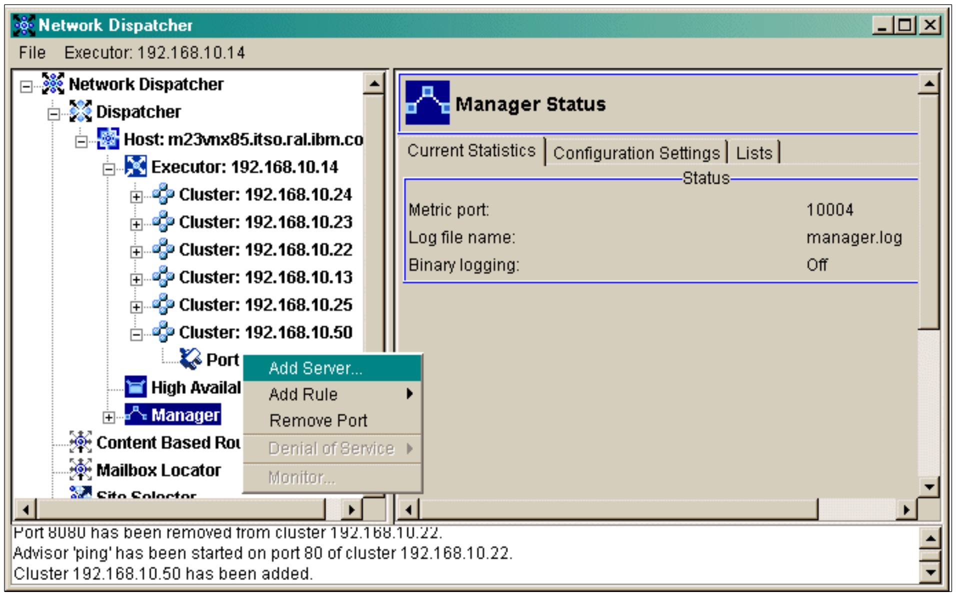


Figure 7-49 Specify the port

5. Specify the port number. In our case, we chose port 80. Leave the forwarding method at `MAC Based forwarding`. Refer to 4.2.2, “Media Access Control (MAC) forwarding” on page 65 for a review of MAC forwarding.

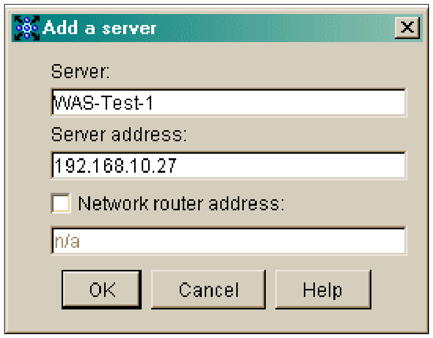


Metric port:	10004
Log file name:	manager.log
Binary logging:	Off

```
Port 8080 has been removed from cluster 192.168.10.22.
Advisor 'ping' has been started on port 80 of cluster 192.168.10.22.
Cluster 192.168.10.50 has been added.
```

Figure 7-50 Add a server to the port

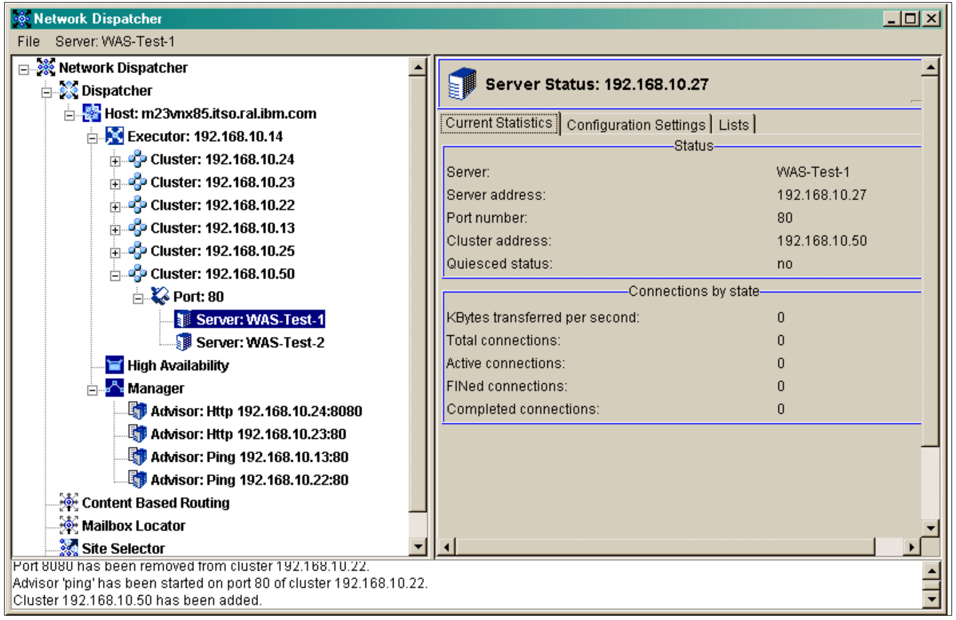
6. Now we add the load balanced back-end application servers to the port. Right-click the port item you created and select **Add Server**.



Server:
WAS-Test-1
Server address:
192.168.10.27
Network router address:
n/a

Figure 7-51 Server data

7. Specify a server name in the Add a server window. It can be the hostname of the server or any display name. We chose to use descriptive names. If you do so, you have to specify the IP address of the server. Otherwise, the GUI will fill it in for you. Leave the Network router box unchecked and do not fill in an address in the bottom text field.



Status

Server:	WAS-Test-1
Server address:	192.168.10.27
Port number:	80
Cluster address:	192.168.10.50
Quiesced status:	no

Connections by state

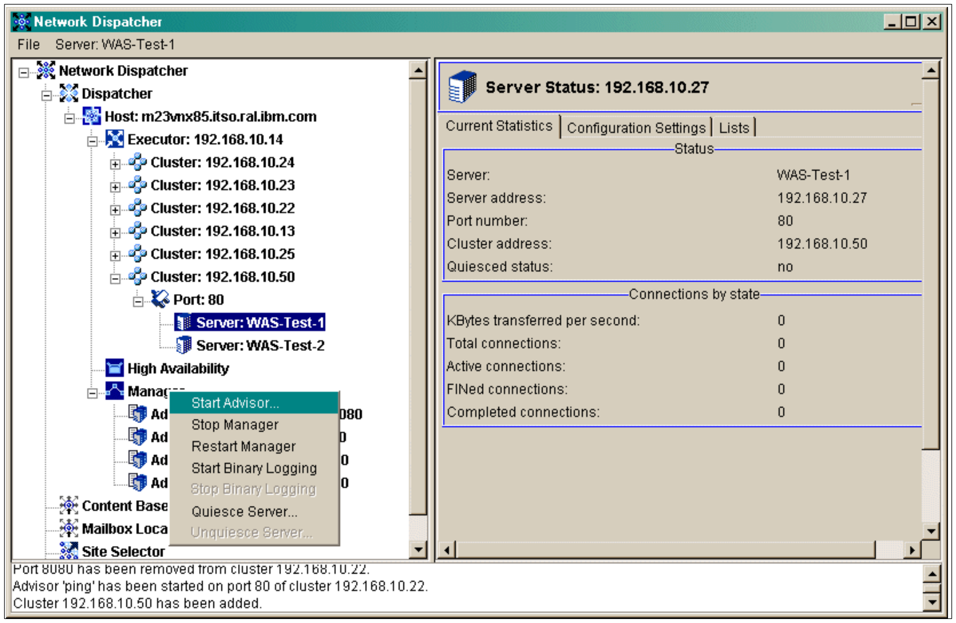
KBytes transferred per second:	0
Total connections:	0
Active connections:	0
FINed connections:	0
Completed connections:	0

```
Port 8080 has been removed from cluster 192.168.10.22.
Advisor 'ping' has been started on port 80 of cluster 192.168.10.22.
Cluster 192.168.10.50 has been added.
```

Figure 7-52 Server status

8. In Figure 7-52, you can see our cluster containing port 80 and two servers. If you have not done so yet, right-click the **Executor** item and select **Start Manager**.

9. To have more accurate measurements to distribute the load between the servers, we start an advisor to monitor the servers in the cluster.



Status

Server:	WAS-Test-1
Server address:	192.168.10.27
Port number:	80
Cluster address:	192.168.10.50
Quiesced status:	no

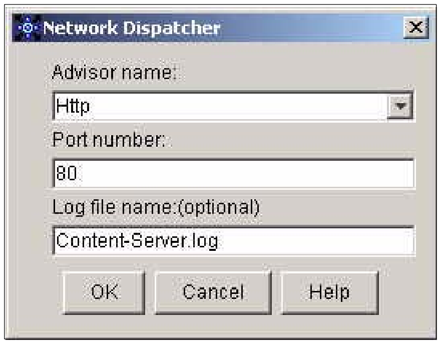
Connections by state

KBytes transferred per second:	0
Total connections:	0
Active connections:	0
FINed connections:	0
Completed connections:	0

```
Port 8080 has been removed from cluster 192.168.10.22.
Advisor 'ping' has been started on port 80 of cluster 192.168.10.22.
Cluster 192.168.10.50 has been added.
```

Figure 7-53 Start advisor

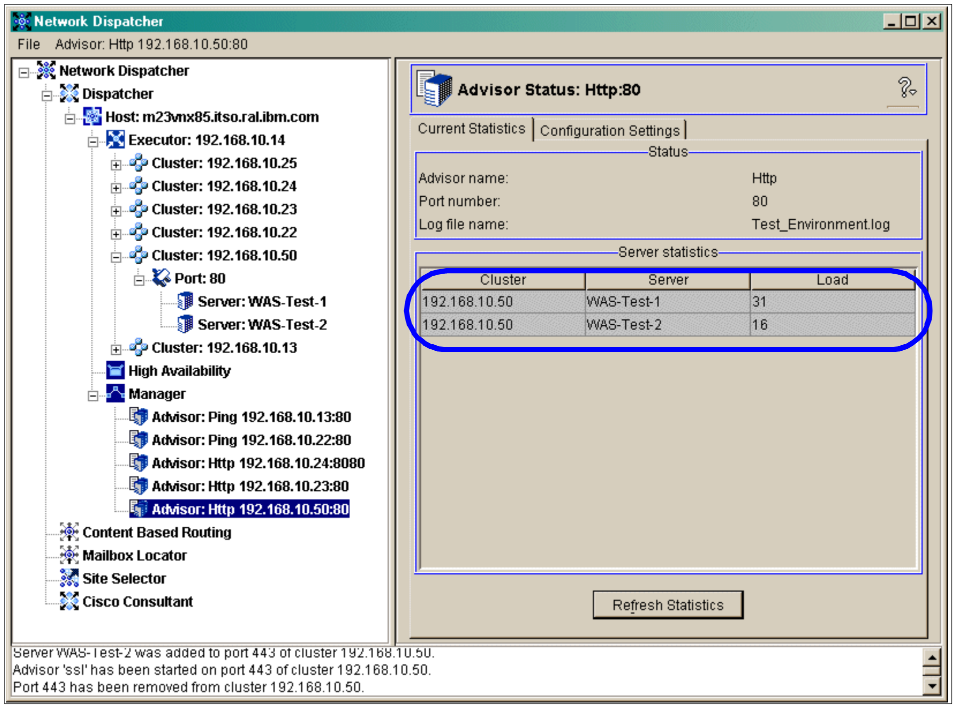
10. Right-click the **Manager** item and select **Start Advisor**.



Advisor name: Http
Port number: 80
Log file name:(optional) Content-Server.log

Figure 7-54 Specify advisor properties

11. Select the HTTP Advisor from the pull-down menu and specify the cluster's port number. Optionally, specify a name for the log file.



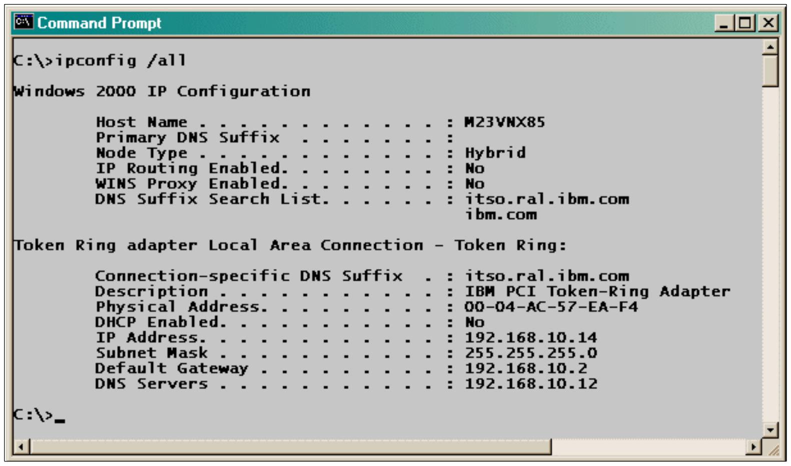
Advisor name:	Http
Port number:	80
Log file name:	Test_Environment.log

Cluster	Server	Load
192.168.10.50	WAS-Test-1	31
192.168.10.50	WAS-Test-2	16

```
Server WAS-Test-2 was added to port 443 of cluster 192.168.10.50.
Advisor 'ssl' has been started on port 443 of cluster 192.168.10.50.
Port 443 has been removed from cluster 192.168.10.50.
```

Figure 7-55 Advisor statistics

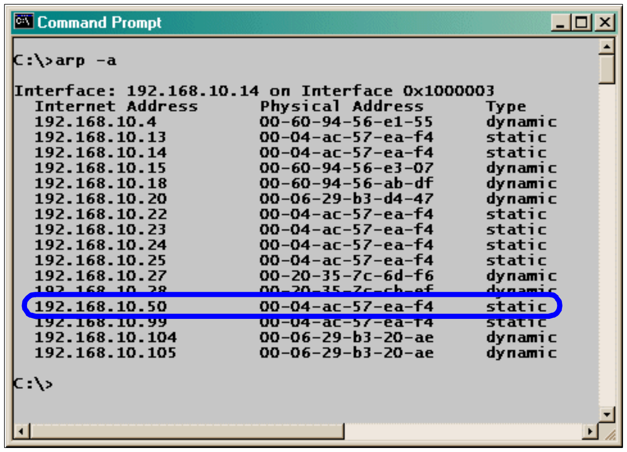
12. After a moment, the advisor will show statistics for each of the clusters' server.



```
C:\>ipconfig /all

Windows 2000 IP Configuration

        Host Name . . . . . . . . . . . . : M23VNX85
        Primary DNS Suffix  . . . . . . . :
        Node Type . . . . . . . . . . . . : Hybrid
        IP Routing Enabled. . . . . . . . : No
        WINS Proxy Enabled. . . . . . . . : No
        DNS Suffix Search List. . . . . . : itso.ral.ibm.com
                                            ibm.com

Token Ring adapter Local Area Connection - Token Ring:

        Connection-specific DNS Suffix  . : itso.ral.ibm.com
        Description . . . . . . . . . . . : IBM PCI Token-Ring Adapter
        Physical Address. . . . . . . . . : 00-04-AC-57-EA-F4
        DHCP Enabled. . . . . . . . . . . : No
        IP Address. . . . . . . . . . . . : 192.168.10.14
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.10.2
        DNS Servers . . . . . . . . . . . : 192.168.10.12

C:\>_
```

Figure 7-56 Output of ipconfig

13. If you run an **`ipconfig /all`** command from the Windows command line, you might wonder why there is not an alias shown for the cluster address. In Windows, it is a special case. The aliases are not really added as an IP address, but added to the ARP table instead. On all Unix and Linux platforms, the **`ifconfig`** command will show the alias on the interfaces.



```
C:\>arp -a

Interface: 192.168.10.14 on Interface 0x1000003
  Internet Address      Physical Address      Type
  192.168.10.4          00-60-94-56-e1-55     dynamic
  192.168.10.13         00-04-ac-57-ea-f4     static
  192.168.10.14         00-04-ac-57-ea-f4     static
  192.168.10.15         00-60-94-56-e3-07     dynamic
  192.168.10.18         00-60-94-56-ab-df     dynamic
  192.168.10.20         00-06-29-b3-d4-47     dynamic
  192.168.10.22         00-04-ac-57-ea-f4     static
  192.168.10.23         00-04-ac-57-ea-f4     static
  192.168.10.24         00-04-ac-57-ea-f4     static
  192.168.10.25         00-04-ac-57-ea-f4     static
  192.168.10.27         00-20-35-7c-6d-f6     dynamic
  192.168.10.28         00-20-35-7c-cb-ef     dynamic
  192.168.10.50         00-04-ac-57-ea-f4     static
  192.168.10.99         00-04-ac-57-ea-f4     static
  192.168.10.104        00-06-29-b3-20-ae     dynamic
  192.168.10.105        00-06-29-b3-20-ae     dynamic

C:\>
```

Figure 7-57 ARP table on Windows

Note: In Windows, the cluster IP address is not aliased to the physical interface, but added as an entry in the ARP table. Therefore, you cannot ping the cluster address from the dispatcher machine itself!

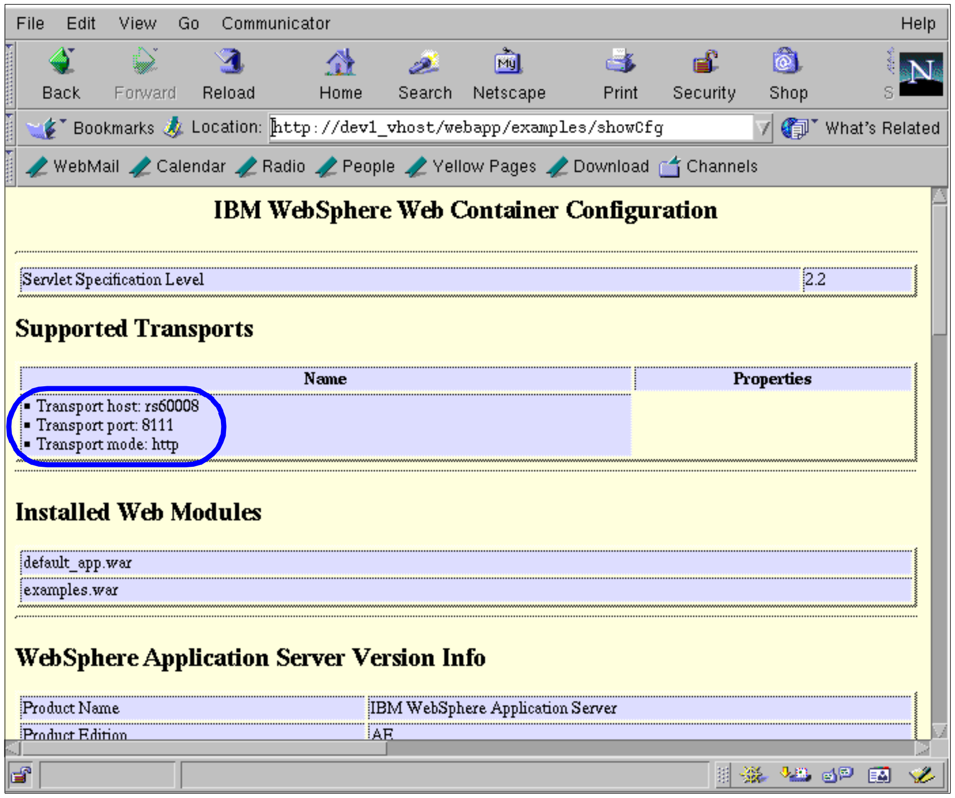
7.2.4 Testing the environment

With the tests we conducted, we wanted to prove two things. First, that it is possible to have the same application installed multiple times in the same environment. Second, that these different instances are independent and do not interfere. Also, we want to show that affinity works within each environment. To test this, we use the showCfg, HitCount and BeenThere servlets of the sample application. This will show that name-based virtual hosts are a very simple technique, yet one that works very well for this purpose.

The environment of developer 1

1. First, we tested the environment of dev1. We opened a browser and pointed to the int_vhost. We entered the following address:

```
http://dev1_vhost/webapp/examples/showCfg
```


File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop
Bookmarks Location: http://dev1_vhost/webapp/examples/showCfg What's Related
WebMail Calendar Radio People Yellow Pages Download Channels

IBM WebSphere Web Container Configuration

Servlet Specification Level	2.2

Supported Transports

Name	Properties
▪ Transport host: rs60008 ▪ Transport port: 8111 ▪ Transport mode: http	

Installed Web Modules

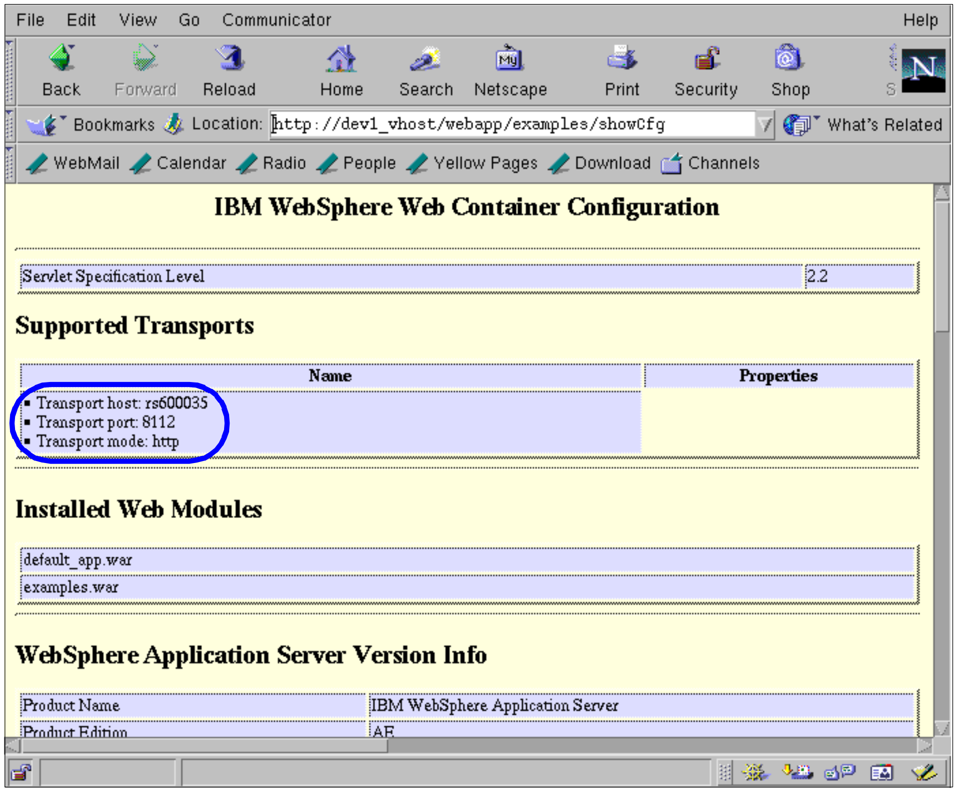
default_app.war
examples.war

WebSphere Application Server Version Info

Product Name	IBM WebSphere Application Server
Product Edition	AE

Figure 7-58 ShowCfg servlet of dev1_vhost

2. Figure 7-58 shows the output of the showCfg servlet.


File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop
Bookmarks Location: http://dev1_vhost/webapp/examples/showCfg
What's Related
WebMail Calendar Radio People Yellow Pages Download Channels

IBM WebSphere Web Container Configuration

Servlet Specification Level	2.2

Supported Transports

Name	Properties
▪ Transport host: rs600035 ▪ Transport port: 8112 ▪ Transport mode: http	

Installed Web Modules

default_app.war
examples.war

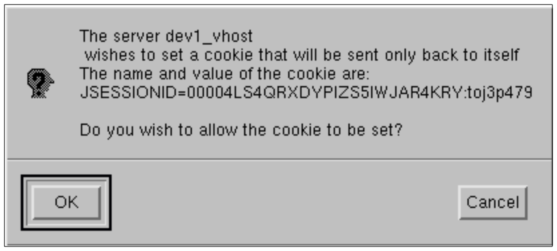
WebSphere Application Server Version Info

Product Name	IBM WebSphere Application Server
Product Edition	AE

Figure 7-59 The reloaded showCfg servlet

3. Figure 7-59 shows that on the second request, the clone on the other node was selected for execution.
4. Next, we entered the address for the HitCount bean page.

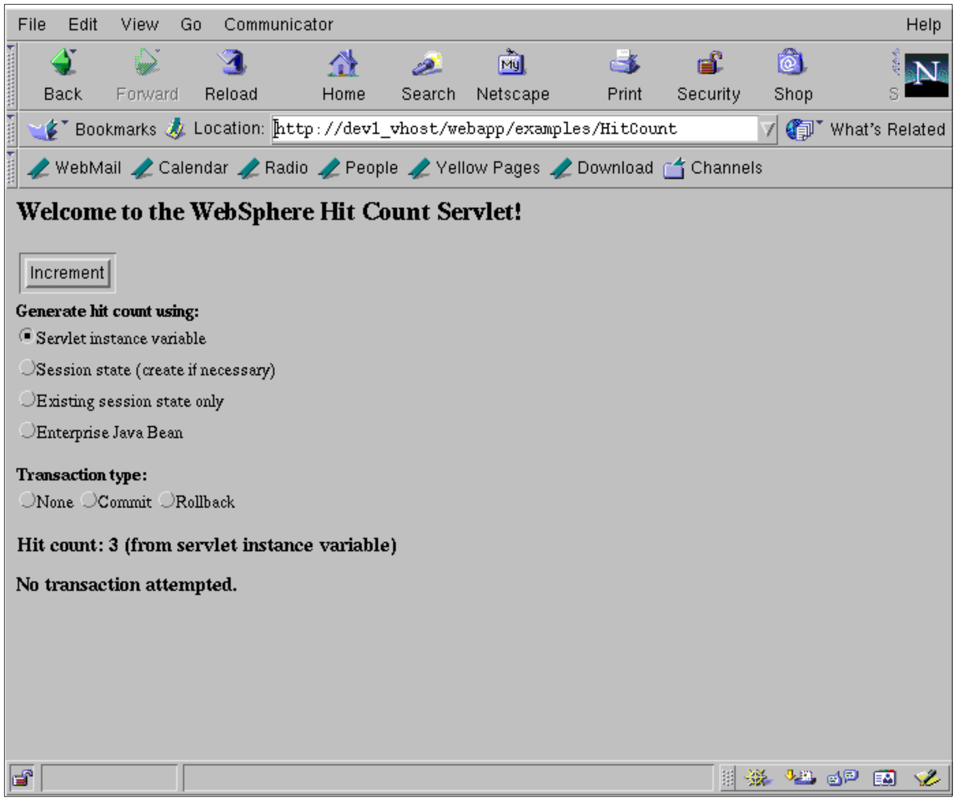
   `http://int_vhost-webapp/examples/HitCount`



```
The server dev1_vhost
 wishes to set a cookie that will be sent only back to itself
The name and value of the cookie are:
JSESSIONID=00004LS4QRXDYPIZS5IWJAR4KRY:toj3p479

Do you wish to allow the cookie to be set?
```

Figure 7-60 Cookie request

5. A cookie was sent to the client and showed it was coming from a clone with the ID toj3p479.


File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop
Bookmarks Location: http://dev1_vhost/webapp/examples/HitCount What's Related
WebMail Calendar Radio People Yellow Pages Download Channels
Increment

Welcome to the WebSphere Hit Count Servlet!

Generate hit count using:

[✓ Servlet instance variable]
Session state (create if necessary)
Existing session state only
Enterprise Java Bean

Transaction type:

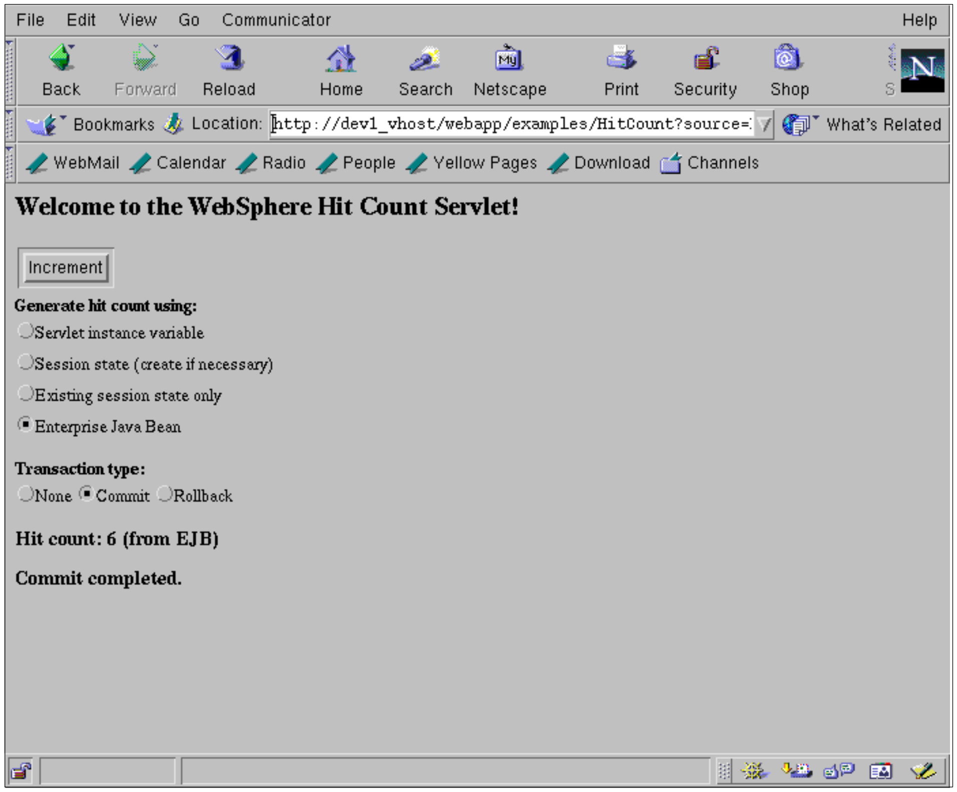
None Commit Rollback

Hit count: 3 (from servlet instance variable)

No transaction attempted.

Figure 7-61 The HitCount servlet page

6. After accepting the cookie, we received the HitCount page. We incremented the session instance variable until the counter showed 3.



http://dev1_vhost/webapp/examples/HitCount?source=

Welcome to the WebSphere Hit Count Servlet!

Increment

Generate hit count using:

Servlet instance variable

Session state (create if necessary)

Existing session state only

[✓ Enterprise Java Bean]

Transaction type:

None [✓ Commit] Rollback

Hit count: 6 (from EJB)

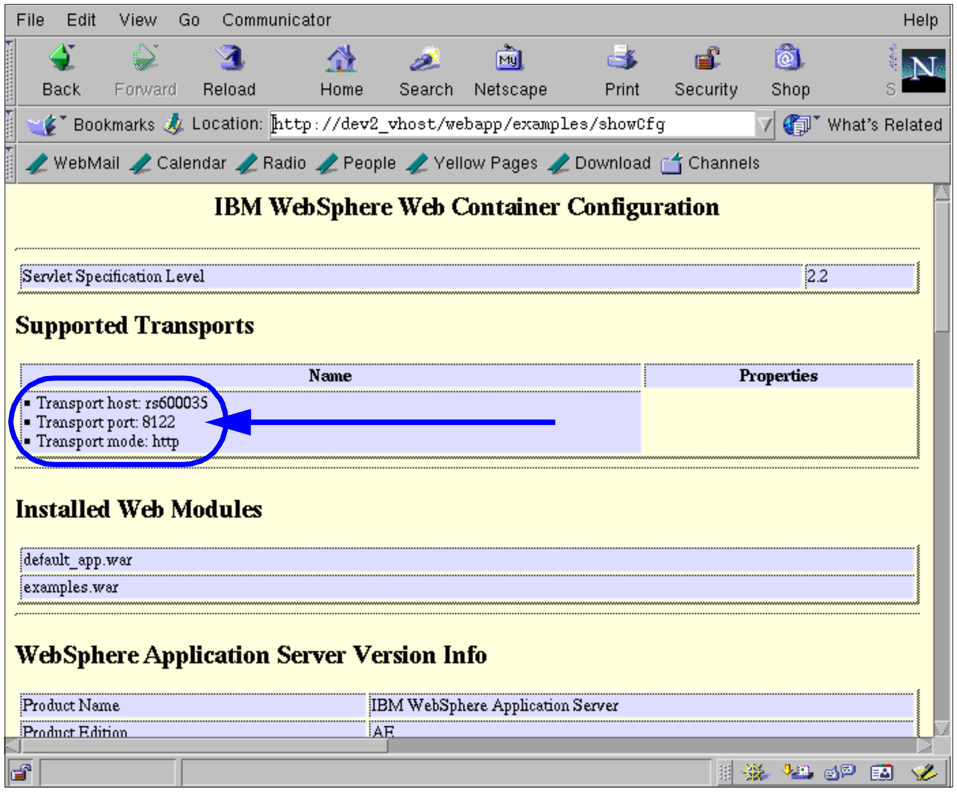
Commit completed.

Figure 7-62 The HitCount page, requesting an EJB increment

7. As the next step, we incremented the Enterprise Java Bean Counter. This is a CMP EJB and this counter is persistent into the INC table of the database.

The environment of developer 2

To test whether the environments are really independent, we did *not* close the browser window. So the cookie we received is still active!



IBM WebSphere Web Container Configuration

Servlet Specification Level	2.2

Supported Transports

Name	Properties
▪ Transport host: rs600035 ▪ Transport port: 8122 ▪ Transport mode: http	

Installed Web Modules

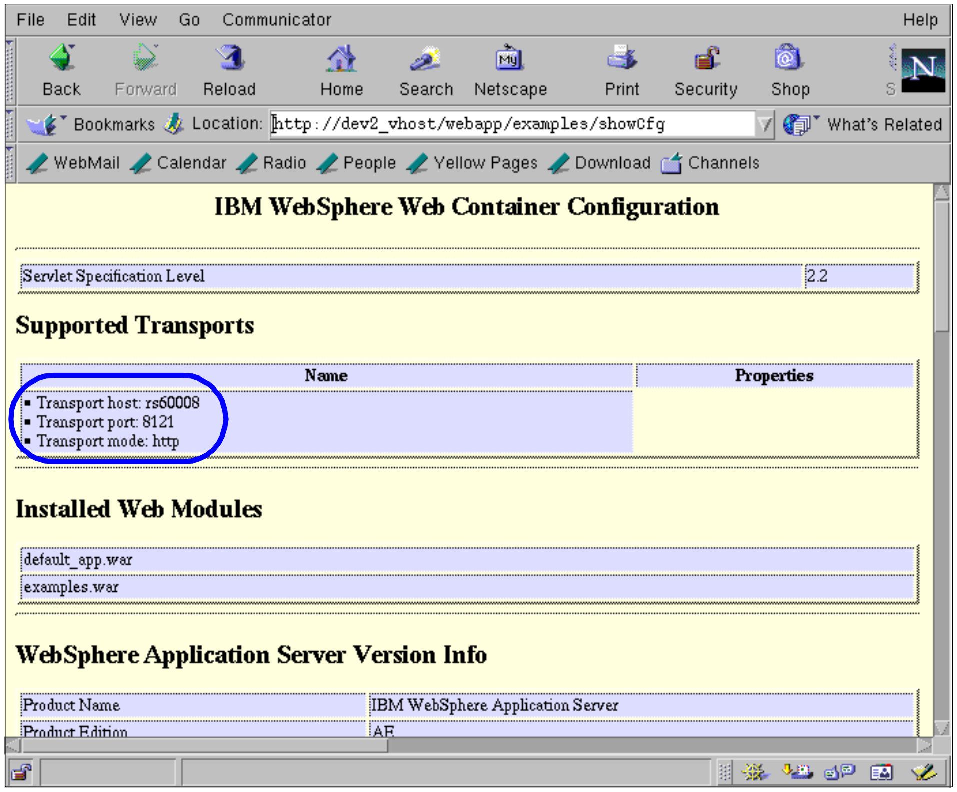
default_app.war
examples.war

WebSphere Application Server Version Info

Product Name	IBM WebSphere Application Server
Product Edition	AE

Figure 7-63 The showCfg servlet of the dev2 environment

1. Now we checked the environment of developer 2, on dev2_vhost. To start, we requested the showCfg servlet again. The port number 8122 proves that it was executed on Clone 2 of the developer 2 environment.


File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop
Bookmarks Location: http://dev2_vhost/webapp/examples/showCfg
What's Related
WebMail Calendar Radio People Yellow Pages Download Channels

IBM WebSphere Web Container Configuration

Servlet Specification Level	2.2

Supported Transports

Name	Properties
▪ Transport host: rs60008 ▪ Transport port: 8121 ▪ Transport mode: http	

Installed Web Modules

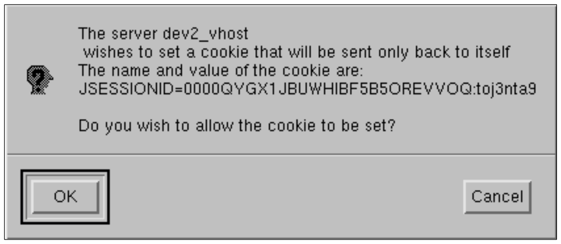
default_app.war
examples.war

WebSphere Application Server Version Info

Product Name	IBM WebSphere Application Server
Product Edition	AE

Figure 7-64 The second request of dev2_host's showCfg servlet

2. A reload shows that the dev2 environment is also balancing the requests.
3. Next, we chose to go to the HitCount page of dev2_host.

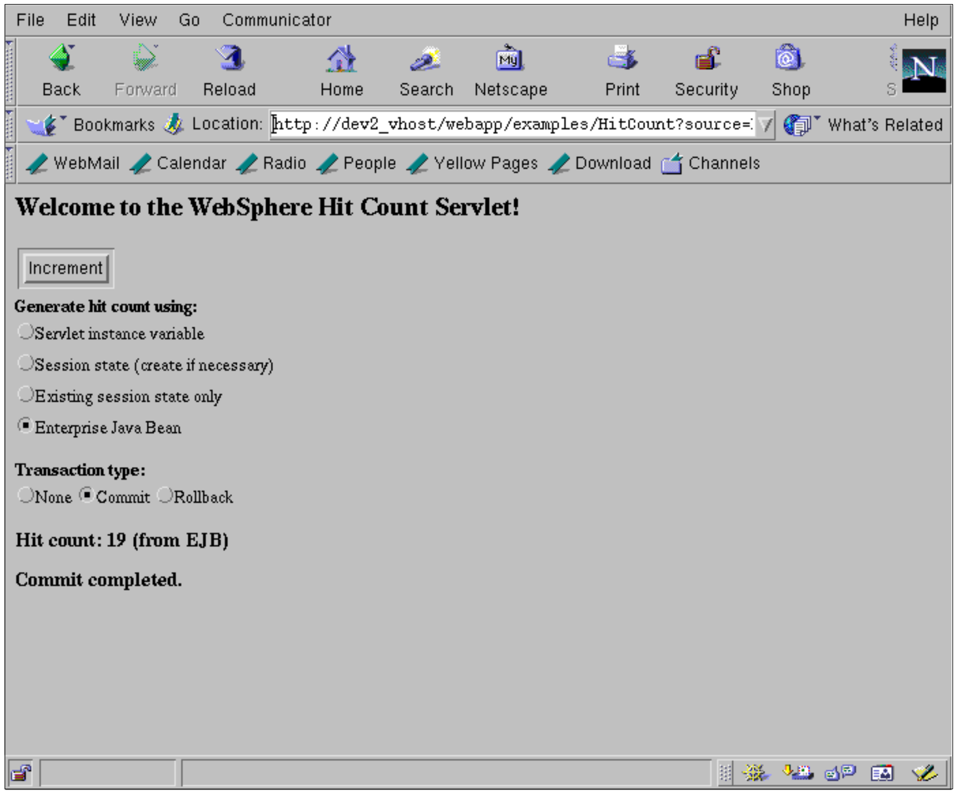


The server dev2_vhost
wishes to set a cookie that will be sent only back to itself
The name and value of the cookie are:
JSESSIONID=0000QYGX1JBUWHIBF5B5OREVVOQ:toj3nta9

Do you wish to allow the cookie to be set?

Figure 7-65 A cookie from dev2_host

4. Another cookie was received, this time from a clone with the ID toj3nta9.



Welcome to the WebSphere Hit Count Servlet!

Increment

Generate hit count using:

Servlet instance variable
Session state (create if necessary)
Existing session state only
[✓ Enterprise Java Bean]

Transaction type:

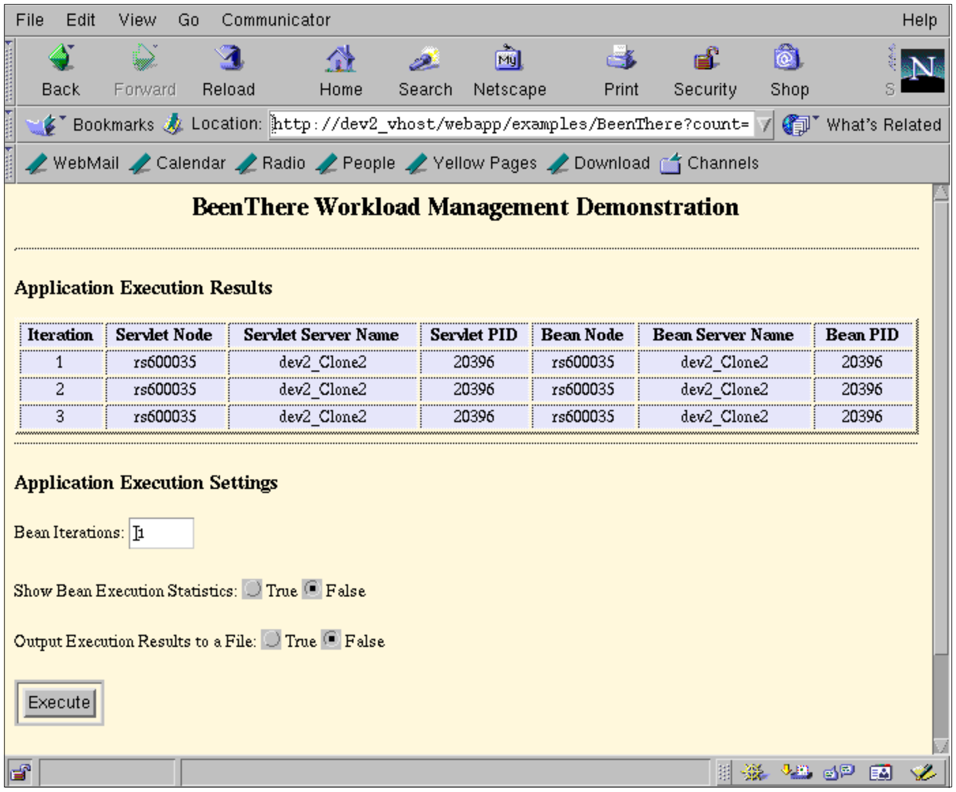
None [✓ Commit] Rollback

Hit count: 19 (from EJB)

Commit completed.

Figure 7-66 The HitCount page of dev2_host

5. On the HitCount page of dev2_host, we incremented to 19. The next page to request was the BeenThere workload management page. We requested three iterations.


File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop
Bookmarks Location: http://dev2_vhost/webapp/examples/BeenThere?count= What's Related
WebMail Calendar Radio People Yellow Pages Download Channels

BeenThere Workload Management Demonstration

Application Execution Results

Iteration	Servlet Node	Servlet Server Name	Servlet PID	Bean Node	Bean Server Name	Bean PID
1	rs600035	dev2_Clone2	20396	rs600035	dev2_Clone2	20396
2	rs600035	dev2_Clone2	20396	rs600035	dev2_Clone2	20396
3	rs600035	dev2_Clone2	20396	rs600035	dev2_Clone2	20396

Application Execution Settings

Bean Iterations: 1

Show Bean Execution Statistics: True [✓ False]

Output Execution Results to a File: True [✓ False]

Figure 7-67 The BeenThere WLM demo page of dev2_host

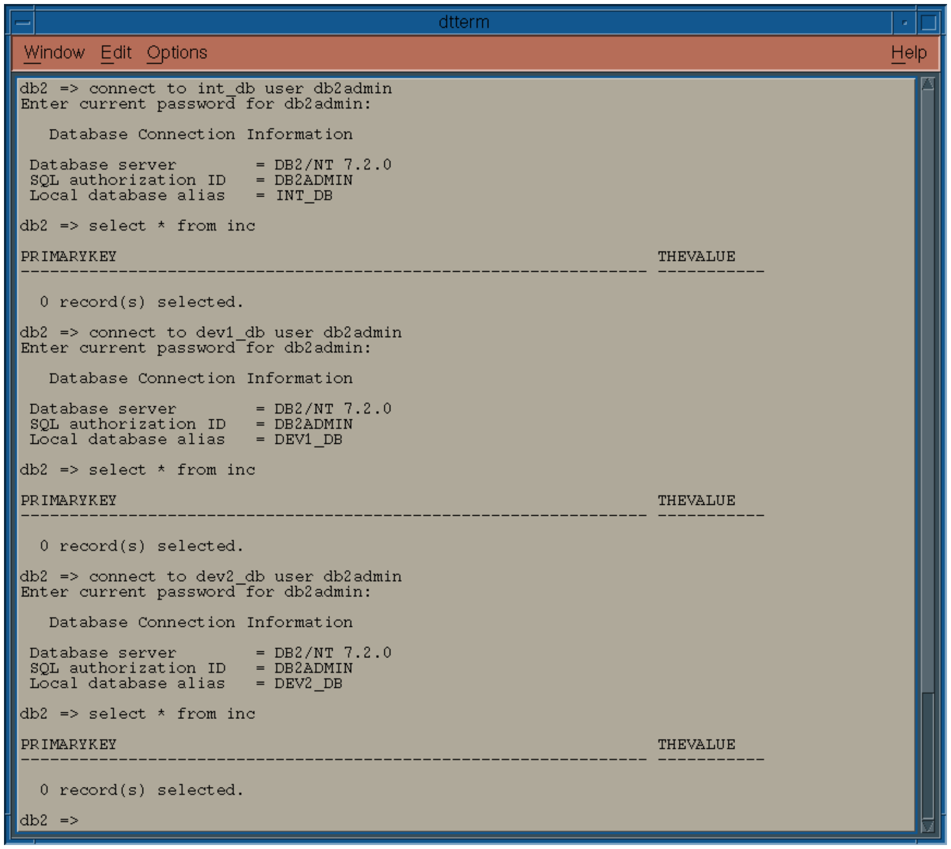
6. As you can see, it was always executed on the same clone, dev2_Clone2, due to the cookie we received when requesting the HitCount page. This shows that the cookie affinity works.

The integration environment

We performed the same tests on the int_vhost environment. We will not show the additional screenshots and will provide a summary only. The showCfg servlet did balance as expected. We incremented the session counter to 5 and the EJB counter to 27. After receiving a cookie for the integration environment, all BeenThere iterations were executed on the same clone.

Summary

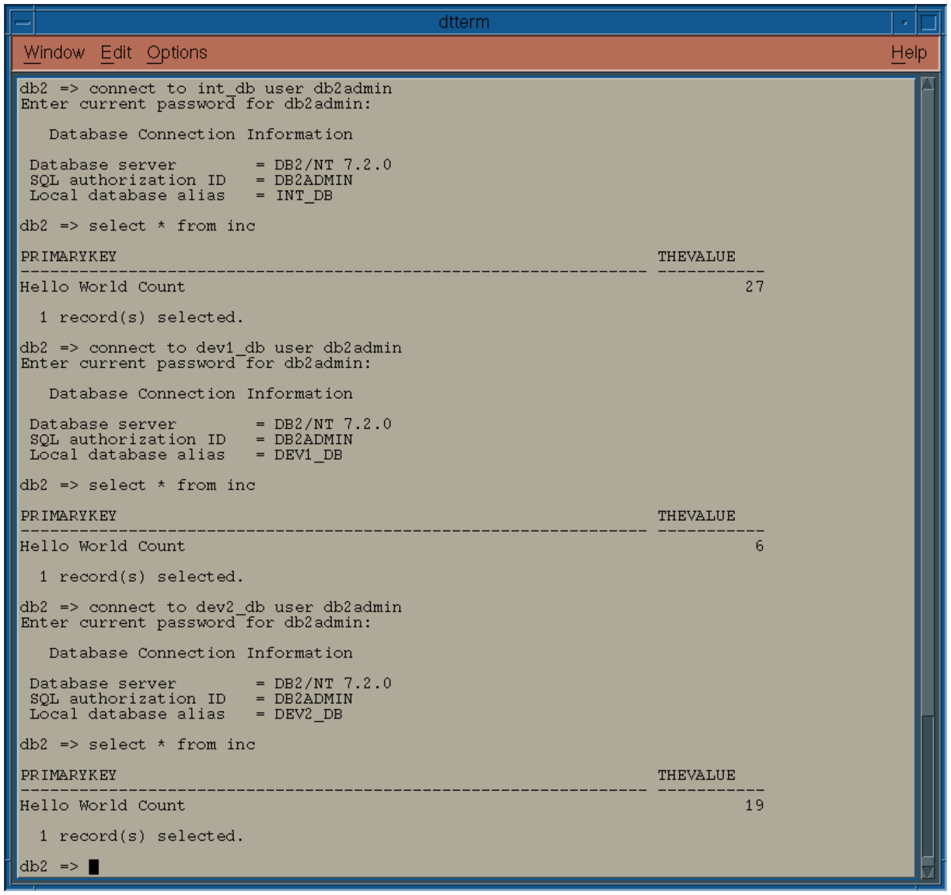
The tests we conducted show clearly the we actually created three environments which are fully independent, yet utilize one IP address only; therefore, all traffic goes through one single cluster. After the tests, we queried the three tables where the EJB counter variable is stored; we show it in one single picture. One was taken before the tests and one afterwards.



```
db2 => connect to int_db user db2admin
Enter current password for db2admin:

   Database Connection Information

 Database server        = DB2/NT 7.2.0
 SQL authorization ID   = DB2ADMIN
 Local database alias   = INT_DB

db2 => select * from inc

PRIMARYKEY                                                        THEVALUE
----------------------------------------------------------------- -----------

  0 record(s) selected.

db2 => connect to dev1_db user db2admin
Enter current password for db2admin:

   Database Connection Information

 Database server        = DB2/NT 7.2.0
 SQL authorization ID   = DB2ADMIN
 Local database alias   = DEV1_DB

db2 => select * from inc

PRIMARYKEY                                                        THEVALUE
----------------------------------------------------------------- -----------

  0 record(s) selected.

db2 => connect to dev2_db user db2admin
Enter current password for db2admin:

   Database Connection Information

 Database server        = DB2/NT 7.2.0
 SQL authorization ID   = DB2ADMIN
 Local database alias   = DEV2_DB

db2 => select * from inc

PRIMARYKEY                                                        THEVALUE
----------------------------------------------------------------- -----------

  0 record(s) selected.

db2 =>
```

Figure 7-68   The three INC tables before the tests

‘



```
db2 => connect to int_db user db2admin
Enter current password for db2admin:

   Database Connection Information

 Database server        = DB2/NT 7.2.0
 SQL authorization ID   = DB2ADMIN
 Local database alias   = INT_DB

db2 => select * from inc

PRIMARYKEY                                                       THEVALUE
---------------------------------------------------------------- -----------
Hello World Count                                                         27

  1 record(s) selected.

db2 => connect to dev1_db user db2admin
Enter current password for db2admin:

   Database Connection Information

 Database server        = DB2/NT 7.2.0
 SQL authorization ID   = DB2ADMIN
 Local database alias   = DEV1_DB

db2 => select * from inc

PRIMARYKEY                                                       THEVALUE
---------------------------------------------------------------- -----------
Hello World Count                                                          6

  1 record(s) selected.

db2 => connect to dev2_db user db2admin
Enter current password for db2admin:

   Database Connection Information

 Database server        = DB2/NT 7.2.0
 SQL authorization ID   = DB2ADMIN
 Local database alias   = DEV2_DB

db2 => select * from inc

PRIMARYKEY                                                       THEVALUE
---------------------------------------------------------------- -----------
Hello World Count                                                         19

  1 record(s) selected.

db2 => █
```

Figure 7-69   The three INC tables after the tests

8

Sample ASP environment

This sample is based on 3.4.3, “Basic Runtime pattern variation 7: caching proxy” on page 52. Though it provides the flexibility to serve identical applications to different host names without interfering, this would be suitable for an Application Service Provider (ASP) environment. It has complete high availability and scalability. Resources can easily be added or shifted. It extends the example Test Environment. To further enhance performance and scalability, a caching proxy layer has been added.

To further improve performance and scalability, we will exploit the possibilities of dynamic caching. This allows caching of the dynamic responses from servlet and JSP calls. This is a technique that will dramatically improve response times for dynamically generated content. At the same time, it improves scalability by freeing resources on the application servers.

The configuration description will guide you not only in how to configure the environment, but also in the configuration and deployment of the sample applications. This will provide you with the possibility to recreate the scenario and experiment with it to acquire confidence for your own project.

Security is not a major concern in this example. We placed all servers in the DMZ. This would be acceptable for public applications with no sensitive data involved. The weather forecast application we show here is an example where security is not a major concern to the data.

8.1 Scenario description

To demonstrate the capabilities of this configuration, we will deploy a sample weather forecast application. It simulates an online weather forecast web site where customers can get forecasts for their geographic area. We created two virtual companies for which we will provide the hosting: www.cloud and www.storm.

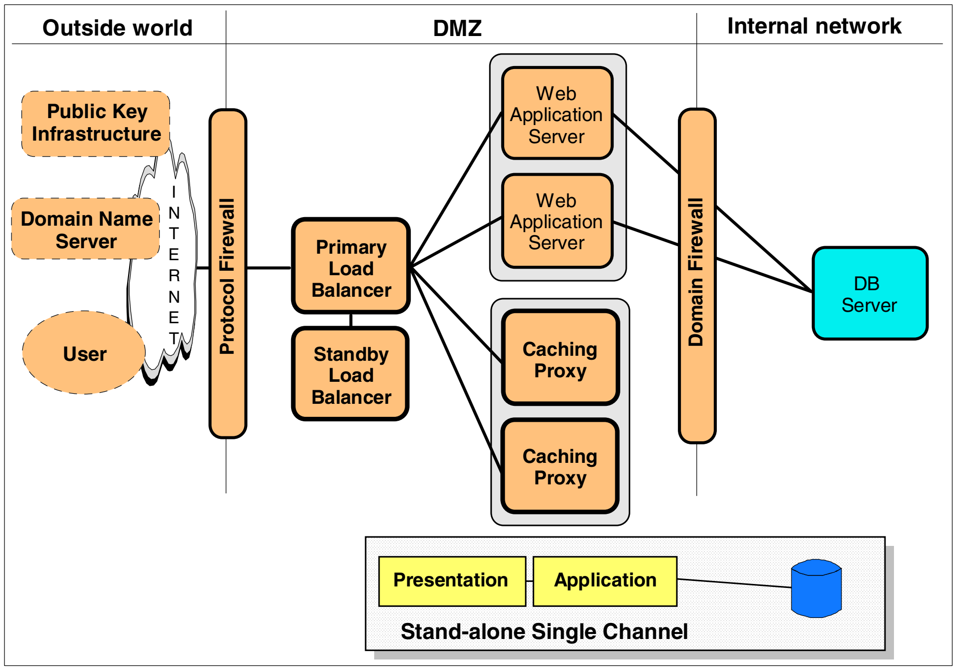


Figure 8-1 Example ASP environment physical topology

Figure 8-1 illustrates the physical configuration of the machines in this scenario. Figure 8-2 illustrates more closely the logical configuration of machines in this scenario.

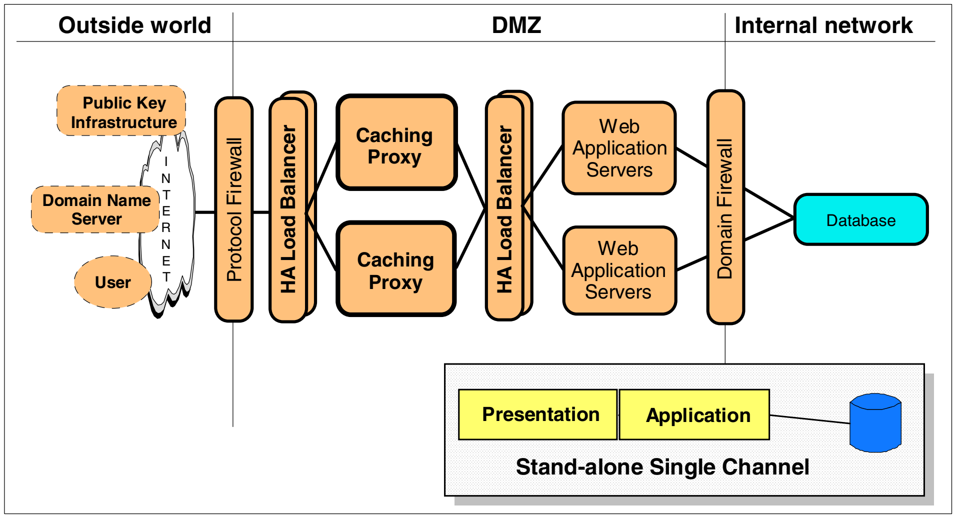


Figure 8-2 Example ASP environment logical topology

The two sets of HA Load Balancer shown in Figure 8-2 are actually the same HA load balancer machines performing load balancing functions for two clusters: the caching proxy cluster and the Web Application Server cluster.

8.1.1 Traffic flow

The following figure and numbered text are used to illustrate request and response traffic flow through the load balancer, caching proxy and Web Application Server machines. The highlighted numbers in the figure correspond to the numbered items in the explanation that follows the figure. The numbers within parenthesis in the figure are the port numbers used for communication between the various machines.

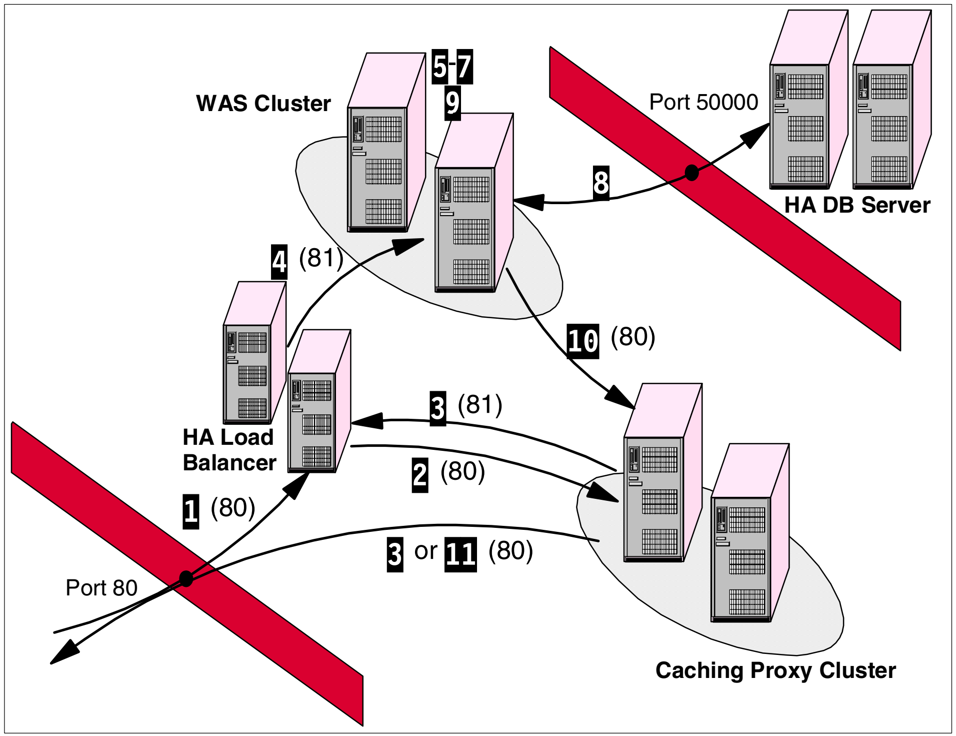


Figure 8-3 Traffic flow

1. The virtual www addresses are all mapped to the same IP cluster address and arrive at the load balancer node. The load balancer listens on port 80 for incoming traffic. It uses the MAC forwarding method and does not examine the contents of the request header or data.
2. The load balancer distributes the requests between the caching proxy nodes without changing the IP address within the packet.
3. The selected caching proxy checks whether the request can be served from cache. If it is in the cache, the response is sent directly back to the client. If it has not been cached,the caching proxy forwards the requests to the content servers. Depending on the hostname contained in the request header, it forwards the request (change the hostname in the request header and the IP address of the packet) to the virtual host address specified by the mapping rule. Because these virtual host names are again mapped to a cluster IP address on the Load Balancer node, they are sent there, but arrive at a different cluster on port 81.
4. The packet does a second turn through the Load Balancer node. This time, the load balancer sends the request to the Web Application Server nodes.

5. The request header now contains the hostname the caching proxy put in. The WebSphere plug-in on the Web Application Server decides, according to a match of this hostname and the requested URI, to which ServerGroup the request belongs and distributes the requests between the clones of this ServerGroup. It changes the IP address of the packet to the one of the transport host for this clone.The hostname in the header remains unchanged. Because each clone on a given host has a unique port number, it will connect to the right one and send the packet.
6. Each clone in turn has its own HTTP daemon, which receives the packet. The Web Container that owns the application context root found in the URI will receive the request.
7. The Web Container determines the right servlet instance to process the request and forwards it for processing.
8. If needed, connections to the database are used to gather information.
9. The servlet response is given back to the HTTP daemon of the clone.
10. The clone's HTTP daemon swaps the source and the destination IP addresses of the packet and sends it back on its way. This is how any server responds to a client.
11. The caching proxy receives the packet and remembers the original client IP address. It puts this IP address as the destination and itself as the source IP address of the packet and sends it back to the client.

Once these two clusters are set up, we can configure as many virtual host names as we like and process them always through the same configuration without any change. All we have to do to set up a new customer is to create a new virtual host on the back-end servers, map them to the existing cluster IP address and add a mapping rule on the proxy server nodes.

8.1.2 Scalability

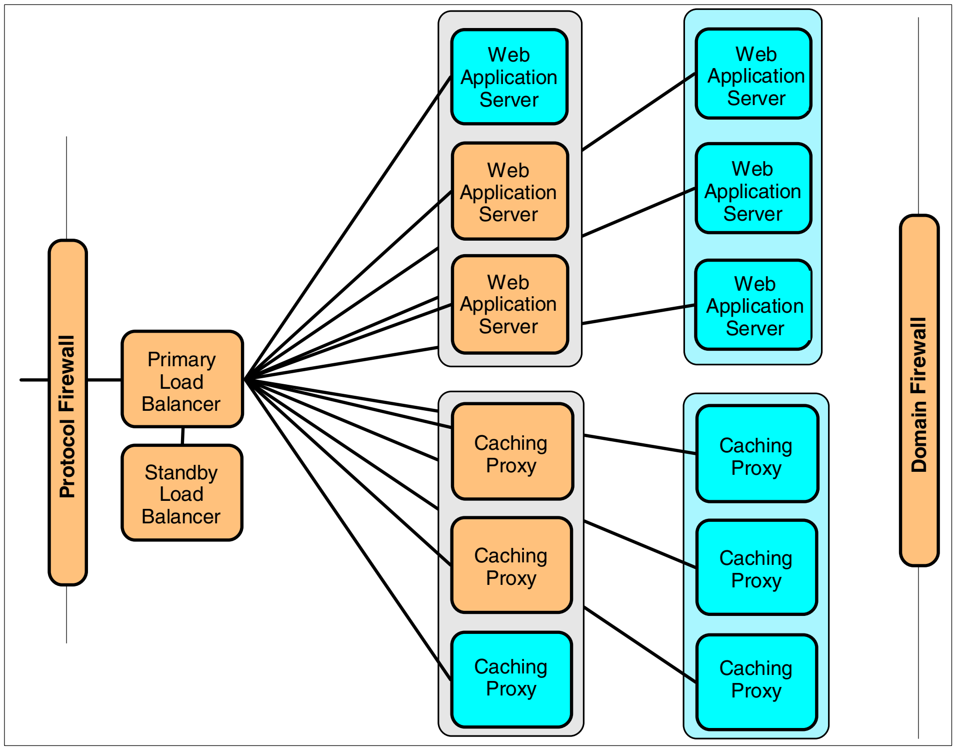


Figure 8-4 Scalability

Full scalability is provided at any level. Servers can be added to any cluster and even whole clusters could be added to extend the configuration. Also, each customer can be scaled to its needs by adding more clones to its ServerGroup or even creating an additional ServerGroup for the customer.

8.1.3 Naming

Note that in this scenario, the virtual hosts again are name-based virtual hosts. They are all based on a single IP address. This is very easy to extend. There is no need for an additional IP address when a new customer is defined. And even more importantly, the load balancer configuration never needs to be changed. Simply add a new definition to DNS or the hosts files.

Table 8-1 Naming of company specific configuration items

Item	cloud	storm
Internet virtual host	www.cloud	www.storm
Cluster IP address	192.168.10.60	192.168.10.60
Back-end virtual host	cloudhost	stormhost
Cluster IP address	192.168.10.70	192.168.10.70

8.1.4 Node functionalities

Table 8-2 Machines used in the test environment scenario

Host Name	IP Address	OS and Software	Service
23VNX85	192.168.10.14	Windows 2000 WebSphere Edge Server V2.0	Primary Load Balancer
23VNX72	192.168.10.15	Windows 2000 WebSphere Edge Server V2.0	Standby Load Balancer
M23BK63Z	192.168.10.18	WebSphere Edge Server V2.0	Caching Proxy
M23BK60H	192.168.10.4	WebSphere Edge Server V2.0	Caching Proxy
RS60008	192.168.10.27	AIX 4.3.3 IBM WebSphere Application Server V4.0.3 IBM HTTP Server V 1.3.19	Web Application Server
RS600035	192.168.10.29	AIX 4.3.3 IBM WebSphere Application Server V4.0.3 IBM HTTP Server V 1.3.19	Web Application Server
M23VNX64	9.24.105.112	Windows 2000 IBM DB2 V7.2	Database Server

The whole scenario can be realized with seven nodes: two load balancers, two caching proxy nodes, two Web Application Server nodes and one database server node. The databases could probably be created on an existing database server.

8.2 Basic configuration description

Following are the basic configuration steps that were performed for the machines used in this scenario (see Table 8-2).

8.2.1 Name resolution

In our sample ASP-environment, we configured the following mappings:

Example 8-1 Host mappings

```
# Sample hosts file for ASP-Environment SG24-6822-00
#==============================================================================

127.0.0.1       localhost

# Server Host Names
#==============================================================================

192.168.10.14   m23vnx85        # Primary Load Balancer
192.168.10.15   m23vnx72        # Standby Load Balancer
192.168.10.18   m23bk63z        # Caching Proxy 1
192.168.10.4    m23bk60h        # Caching Proxy 2
192.168.10.27   rs60008         # Content Server 1
192.168.10.29   rs600035        # Content Server 2
9.24.105.112    m23vnx64        # Backend Database Server

# virtual Host Names
#==============================================================================

192.168.10.60   www.cloud       # Virtual www address for Customer 1 Cluster
192.168.10.60   www.storm       # Virtual www address for Customer 2 Cluster
192.168.10.70   cloudhost       # Virtual address for Content Server Cluster 1
192.168.10.70   stormhost       # Virtual address for Content Server Cluster 2
```

8.2.2 DB Server

1. Log on as the DB2 instance owner.

   ```
   #> su - db2inst1
   ```

2. Create the repository database for the WebSphere domain.

   ```
   #> db2 create db asp-db
   ```

3. Enlarge the application heap size.

```
#> db2 update db cfg for asp-db using applheapsz 256
```

4. Test the connection.

```
#> db2 connect to asp-db user db2inst1
```

8.2.3 DB Client

1. On each WebSphere Application Server node, log on as the db instance owner.

```
#> su - db2inst1
```

2. Catalog the remote db server.

```
#> db2 catalog tcpip node rains remote m23vnx64 server 50000
```

3. Catalog the database.

```
#> db2 catalog db asp-db as asp-db at node rains
```

4. Restart the instance.

```
#> db2stop
#> db2start
```

5. Test the connection.

```
#> db2 connect to asp-db user db2inst1
```

8.2.4 Application servers

The configuration of the Web Application Server nodes is the same as in the test environment example (refer to 7.2.2, “WebSphere Application Server” on page 183), using different names. We assume you have two application servers installed, that you have connected to the repository database (asp-db as described previously in 8.2.3, “DB Client”) and that the initial configuration has been installed. You should have a Default Server on each node with the sample applications installed and the basic resources, such as the JDBC driver, in place. If the properties:

```
install.initial.config
```

and

```
com.ibm.ejs.sm.adminServer.createTables
```

were set to `true` in the `<WAS_ROOT>/bin/admin.config` file, this was automatically done at startup time. Start the Administrative Console to begin the configuration.

Create the virtual hosts

1. The first step is to uninstall the sample applications. Right-click the application and select **Remove**. Repeat this for each application that has been installed.
2. Now right-click the virtual host default_host and select **Remove**.
3. Also remove the default_Server on each node. Now we have a clean domain and are ready to start the new configuration.

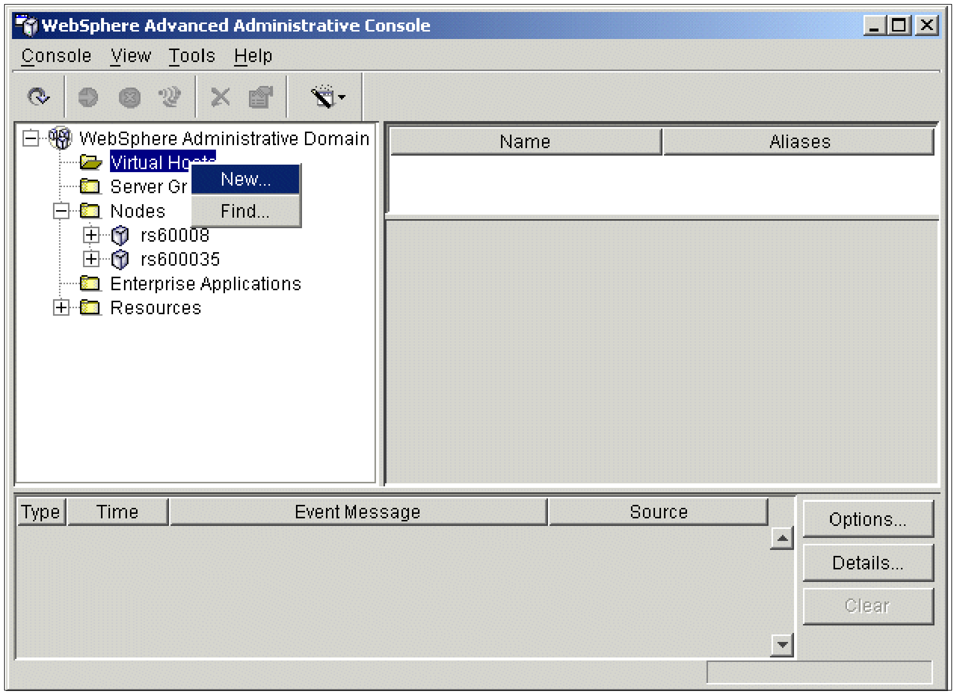
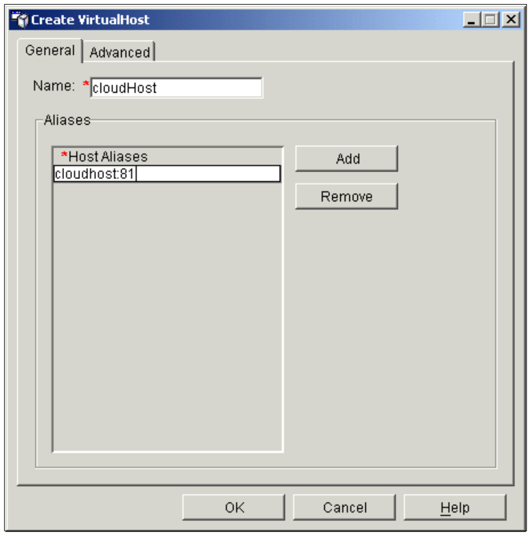


Figure 8-5 Create new virtual host

4. Right-click the **Virtual Host** item in the left-hand pane and select **New** as shown in Figure 8-5.



Name: *cloudHost

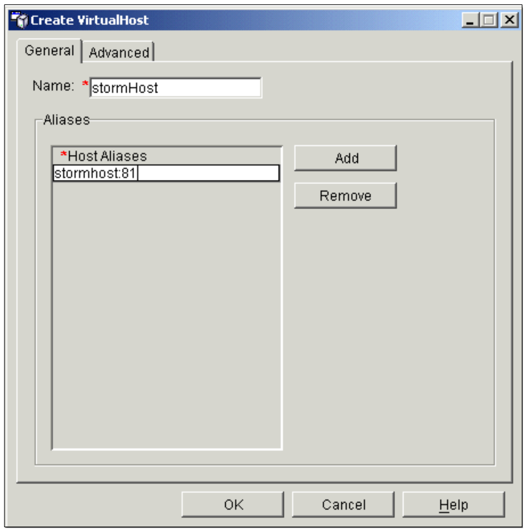
Aliases

*Host Aliases

cloudhost:81

Figure 8-6 The first virtual host definition

5. Specify a name for the new virtual host. We chose to start with the cloudhost definition. Add cloudhost:81 to the alias list. Note that we set it to listen to port 81, as we did for the Web servers.This alias is the virtual hostname we will later use to connect to the application server. Because we are using DNS for the name resolution, we also added the fully qualified hostname as a second entry. Click **OK** to create the virtual host.



Name: *stormHost

Aliases

*Host Aliases

stormhost:81

Figure 8-7 The second virtual host definition

6. .Figure 8-7 shows the second virtual host we created for this environment, the stormhost. Like the first one, it listens on port 81.

Create the ServerGroups

We will now create a ServerGroup for each customer: The Cloud-ServerGroup and Storm-ServerGroup.

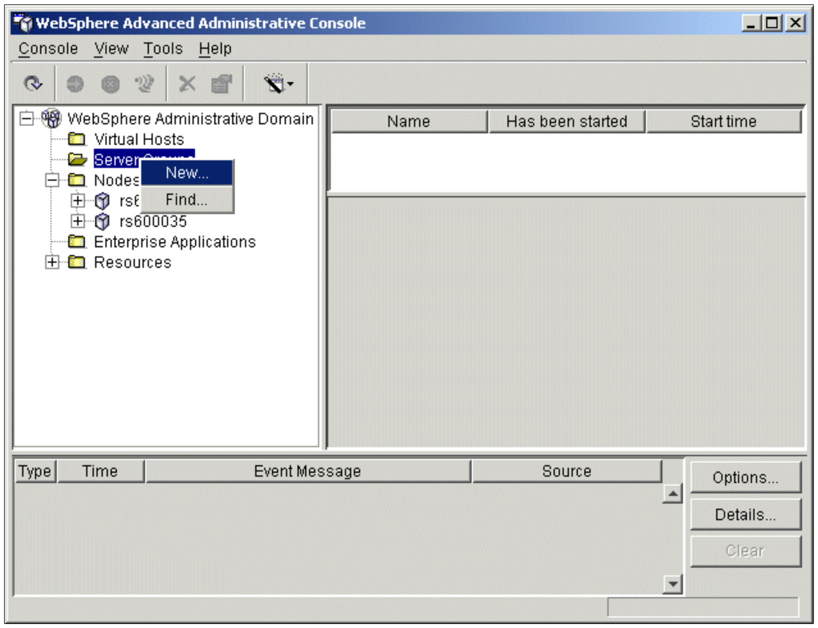
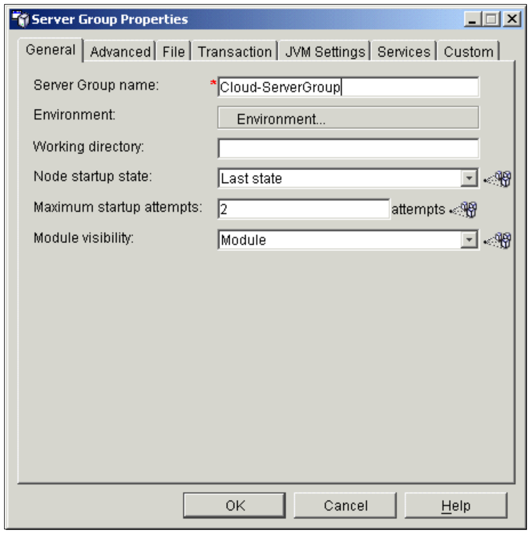


Figure 8-8 Create a ServerGroup

1. In the left-hand pane, right-click the **ServerGroup** item and select **New**.

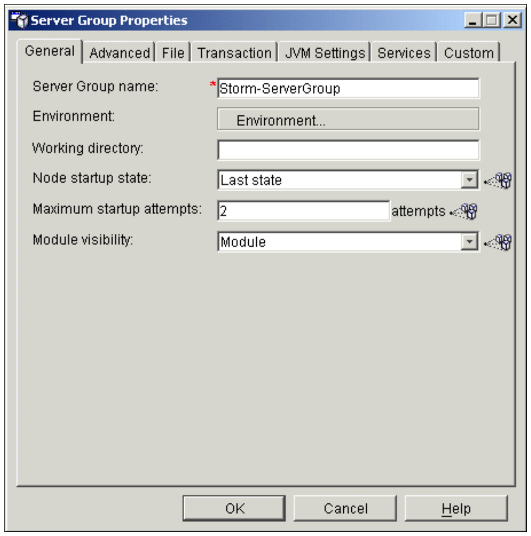


Server Group Properties

Server Group name: *Cloud-ServerGroup
Environment: Environment...
Working directory:
Node startup state: Last state
Maximum startup attempts: 2 attempts
Module visibility: Module

Figure 8-9 Specify ServerGroup properties

2. Specify the name of the new ServerGroup. We started with the Cloud-ServerGroup. Click **OK** to create the ServerGroup.



Server Group Properties

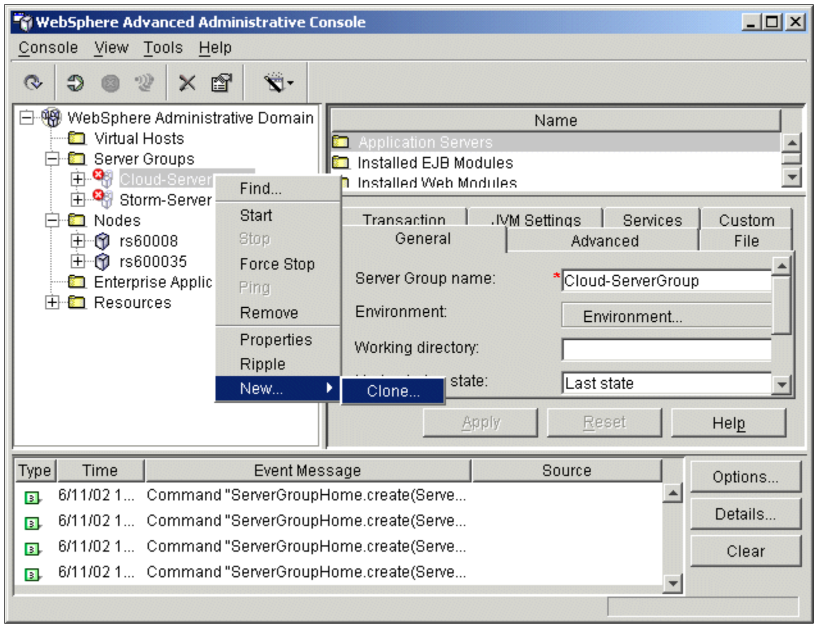
Server Group name: *Storm-ServerGroup
Environment: Environment...
Working directory:
Node startup state: Last state
Maximum startup attempts: 2 attempts
Module visibility: Module

Figure 8-10 Create the second ServerGroup

3. Repeat the procedure to create the Storm-ServerGroup.

Create the clones

In this section, we create two clones to each ServerGroup we created previously.

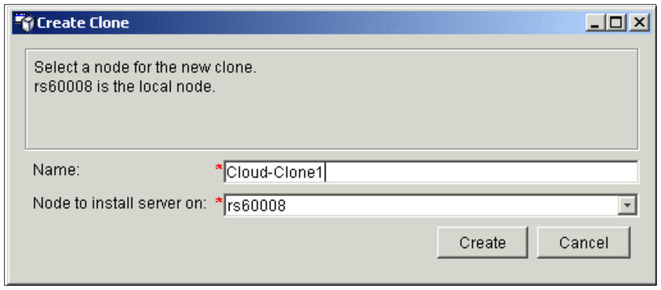


Server Group name: Cloud-ServerGroup
Environment: Environment...
Working directory:
state: Last state

Type	Time	Event Message	Source
	6/11/02 1...	Command "ServerGroupHome.create(Serve...	
	6/11/02 1...	Command "ServerGroupHome.create(Serve...	
	6/11/02 1...	Command "ServerGroupHome.create(Serve...	
	6/11/02 1...	Command "ServerGroupHome.create(Serve...	

Figure 8-11 Create new clone

1. In the left-hand pane, right-click the ServerGroup item you want to add clones to. Select **New -> Clone**.

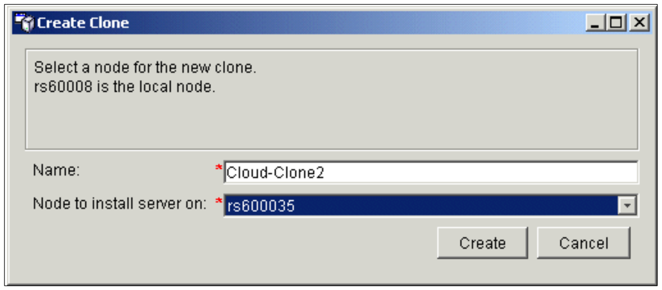


Select a node for the new clone.
rs60008 is the local node.

Name: Cloud-Clone1
Node to install server on: rs60008

Figure 8-12 Specify clone properties

2. Enter the name of the clone. It can be any name, but it has to be unique within this Websphere domain. Also select a node (from the drop-down list) on which to create the clone.
3. Repeat this procedure for the three remaining clones we need.

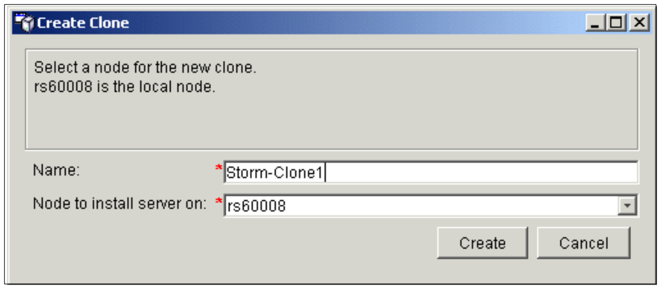


Select a node for the new clone.
rs60008 is the local node.

Name: *Cloud-Clone2

Node to install server on: *rs600035

Figure 8-13 Clone 2 of Cloud company

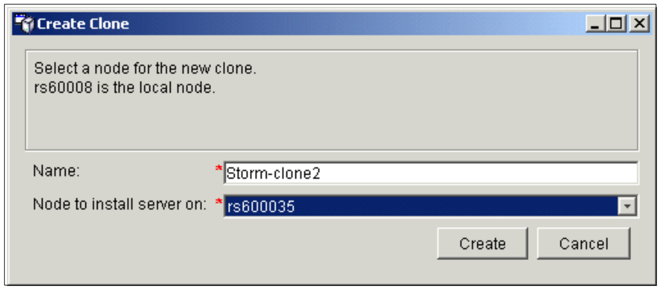


Select a node for the new clone.
rs60008 is the local node.

Name: *Storm-Clone1

Node to install server on: *rs60008

Figure 8-14 Clone 1 of Storm company

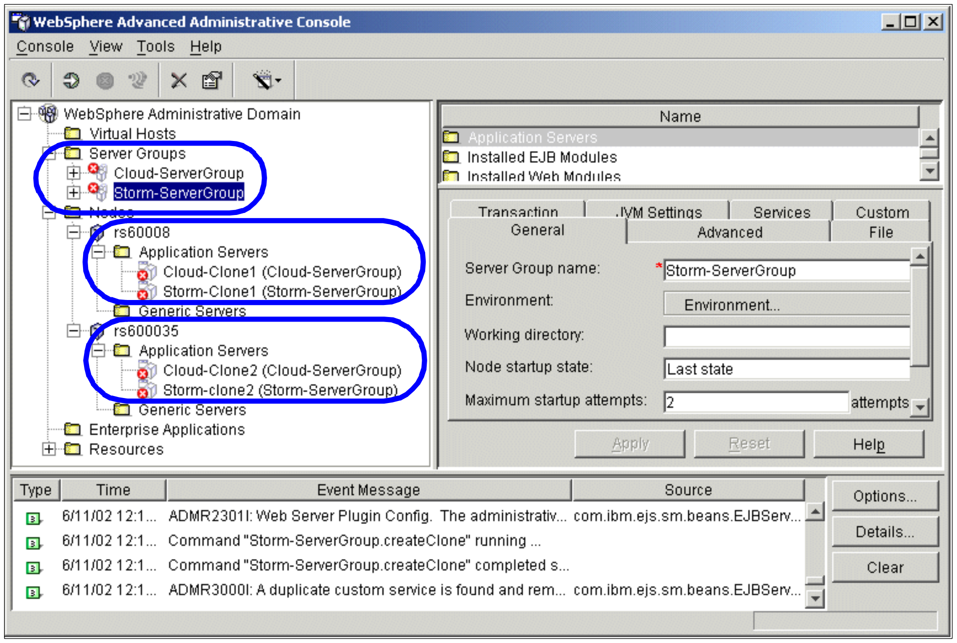


Select a node for the new clone.
rs60008 is the local node.

Name: *Storm-clone2

Node to install server on: *rs600035

Figure 8-15 Clone 2 of Storm company



WebSphere Administrative Domain
- Virtual Hosts
- Server Groups
  - Cloud-ServerGroup
  - Storm-ServerGroup
- Nodes
  - rs60008
    - Application Servers
      - Cloud-Clone1 (Cloud-ServerGroup)
      - Storm-Clone1 (Storm-ServerGroup)
    - Generic Servers
  - rs600035
    - Application Servers
      - Cloud-Clone2 (Cloud-ServerGroup)
      - Storm-clone2 (Storm-ServerGroup)
    - Generic Servers
- Enterprise Applications
- Resources

Name
Application Servers
Installed EJB Modules
Installed Web Modules

Server Group name:	Storm-ServerGroup
Environment:	Environment...
Working directory:	
Node startup state:	Last state
Maximum startup attempts:	2 attempts

Type	Time	Event Message	Source
	6/11/02 12:1...	ADMR2301I: Web Server Plugin Config. The administrativ...	com.ibm.ejs.sm.beans.EJBServ...
	6/11/02 12:1...	Command "Storm-ServerGroup.createClone" running ...	
	6/11/02 12:1...	Command "Storm-ServerGroup.createClone" completed s...	
	6/11/02 12:1...	ADMR3000I: A duplicate custom service is found and rem...	com.ibm.ejs.sm.beans.EJBServ...

Figure 8-16 Overview of the ServerGroups and clones created

4. Figure 8-16 gives an overview of the virtual hosts, ServerGroups and clones we created so far.

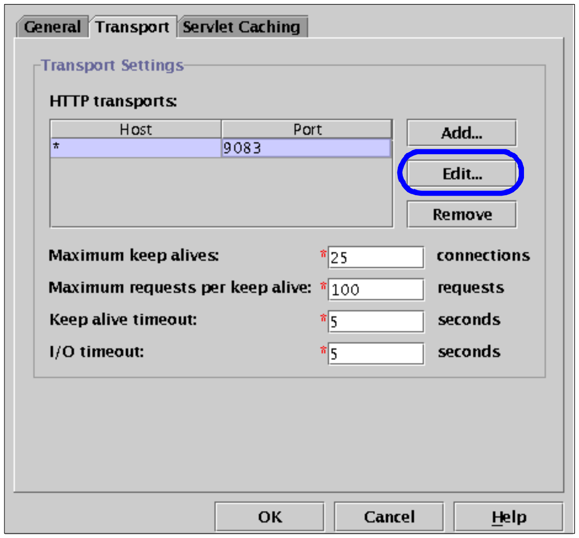
Configure the clones

As in the test environment example, we must configure the clones to use the correct hostnames and to use distinct ports.

Table 8-3 Ports used by the clones

Name	Port	Host
Cloud-Clone1	9081	RS60008
Cloud-Clone2	9082	RS600035
Storm-Clone1	9083	RS60008
Storm-Clone2	9084	RS600035

1. Select a clone in the left-hand pane. Then, in the right-hand pane, select the **Services** tab. A list of services appears within this pane. Select **Web Container Service** and click **Edit Properties.**



Transport Settings

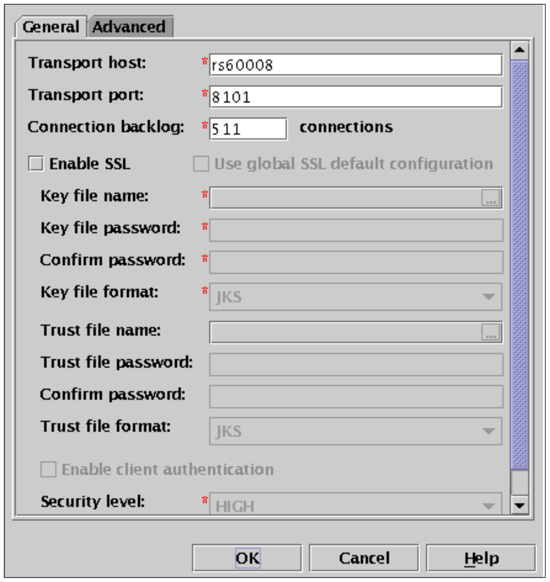
HTTP transports:

Host	Port
*	9083

Maximum keep alives:	* 25	connections
Maximum requests per keep alive:	* 100	requests
Keep alive timeout:	* 5	seconds
I/O timeout:	* 5	seconds

Figure 8-17 The Web container properties

2. The properties window shown in Figure 8-17 appears. First select the **Transport** tab. Then select the **HTTP transport** entry and click **Edit.**



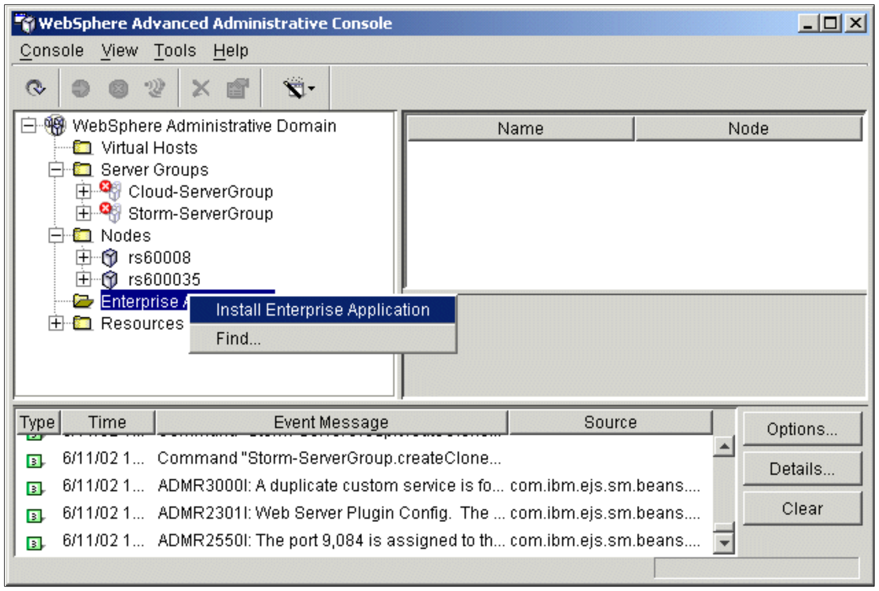
Transport host:	* rs60008
Transport port:	* 8101
Connection backlog:	* 511 connections
☐ Enable SSL	☐ Use global SSL default configuration
Key file name:	*
Key file password:	*
Confirm password:	*
Key file format:	* JKS
Trust file name:	
Trust file password:	
Confirm password:	
Trust file format:	JKS
☐ Enable client authentication	
Security level:	* HIGH

Figure 8-18 Web Container Transport properties

3. Insert the **Transport host**. It is the node name on which the clone resides.

4. Enter the port number to be used by this clone.
5. Click **OK**.
6. Click **Apply** to save the changes.

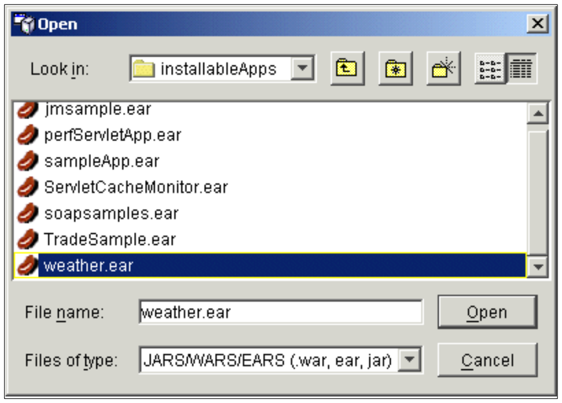
Install the weather application



```
Type  Time        Event Message                                   Source
      6/11/02 1...  Command "Storm-ServerGroup.createClone...
      6/11/02 1...  ADMR3000I: A duplicate custom service is fo...  com.ibm.ejs.sm.beans....
      6/11/02 1...  ADMR2301I: Web Server Plugin Config.  The ...  com.ibm.ejs.sm.beans....
      6/11/02 1...  ADMR2550I: The port 9,084 is assigned to th...  com.ibm.ejs.sm.beans....
```

Figure 8-19 Install Enterprise Application

1. Right-click the **Enterprise Applications** item in the left-hand pane and select **Install Enterprise Application** as shown in Figure 8-19.



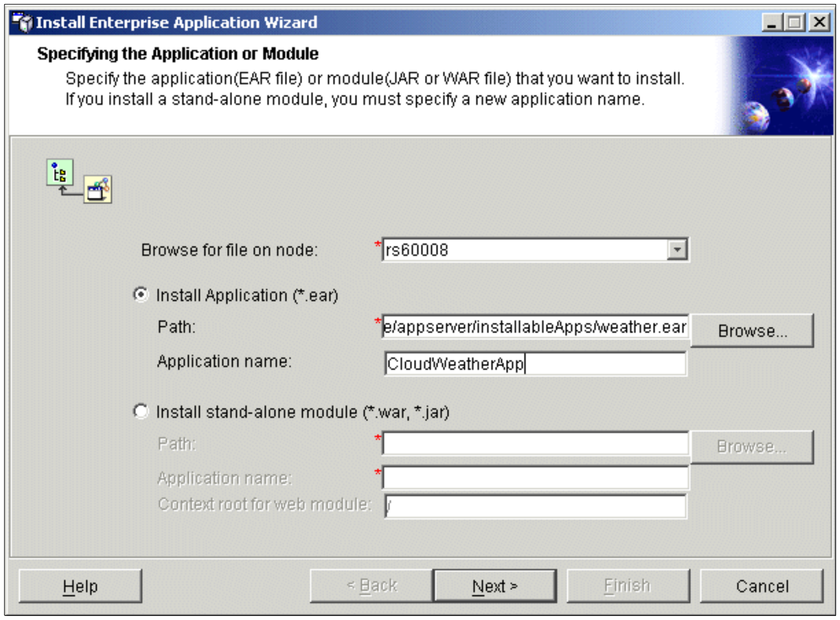
jmsample.ear
perfServletApp.ear
sampleApp.ear
ServletCacheMonitor.ear
soapsamples.ear
TradeSample.ear
weather.ear

File name: weather.ear

Files of type: JARS/WARS/EARS (.war, ear, jar)

Figure 8-20 Browse for an Application file

2. In the Specify Application Module window, click **Browse**. Go to the directory in which you placed the weather.ear file and select it. Click **Open**.


Install Enterprise Application Wizard

Specifying the Application or Module

Specify the application(EAR file) or module(JAR or WAR file) that you want to install.
If you install a stand-alone module, you must specify a new application name.

Browse for file on node: * rs60008

(●) Install Application (*.ear)

Path: * e/appserver/installableApps/weather.ear [Browse...]

Application name: CloudWeatherApp

( ) Install stand-alone module (*.war, *.jar)

Path: *

Application name: *

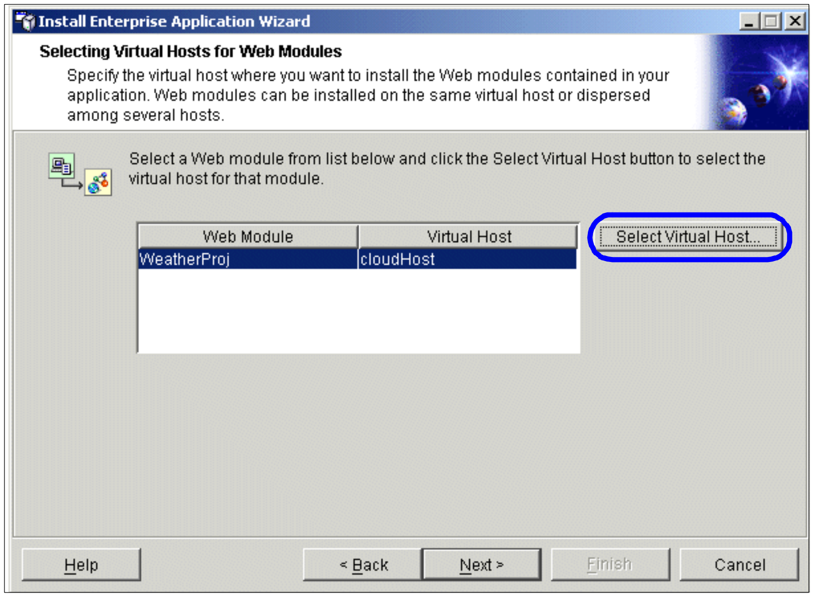
Context root for web module: /

Help | < Back | Next > | Finish | Cancel

Figure 8-21   Specify Application Module window

3. Back in the Module window, specify a name for the application. We chose CloudWeatherApp. Click **Next** to proceed.

4. Go through the next windows and accept the settings until the following window appears.


Install Enterprise Application Wizard

Selecting Virtual Hosts for Web Modules

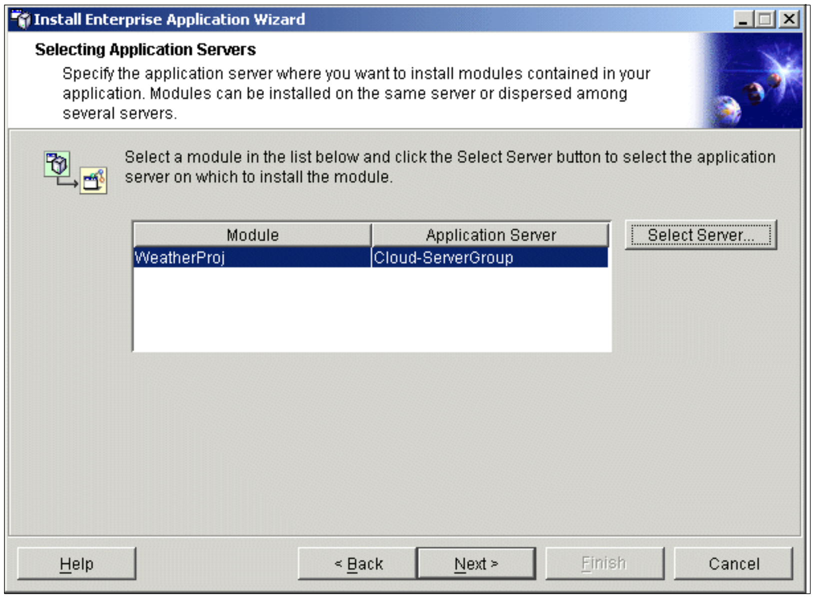
Specify the virtual host where you want to install the Web modules contained in your application. Web modules can be installed on the same virtual host or dispersed among several hosts.

Select a Web module from list below and click the Select Virtual Host button to select the virtual host for that module.

Web Module	Virtual Host
WeatherProj	cloudHost

Figure 8-22 Virtual Host selection window

5. Select the module in the list and click **Select Virtual Host**.
6. Choose the virtual host to be CloudHost.


Install Enterprise Application Wizard

Selecting Application Servers

Specify the application server where you want to install modules contained in your application. Modules can be installed on the same server or dispersed among several servers.

Select a module in the list below and click the Select Server button to select the application server on which to install the module.

Module	Application Server
WeatherProj	Cloud-ServerGroup

Figure 8-23 Application Server selection window

7. Now you must define on which application server (or ServerGroup) the module will run. Select the **Cloud-ServerGroup** as shown in Figure 8-23.
8. In the next window, you will see the directory where the application will be installed. Click **Finish** to start the installation process.

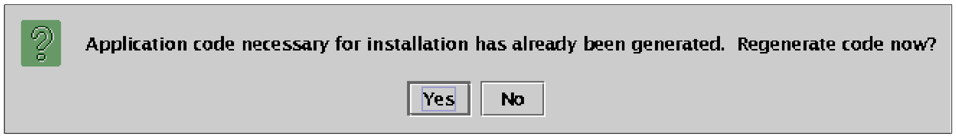

Application code necessary for installation has already been generated. Regenerate code now?

Figure 8-24 Code regeneration window

9. Before it starts, another window appears and allows you to regenerate the code. You do not need to regenerate the code. Click **NO**.

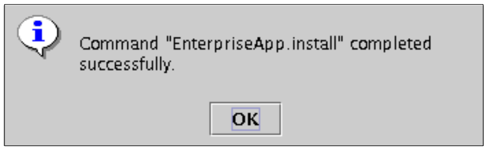

Command "EnterpriseApp.install" completed successfully.

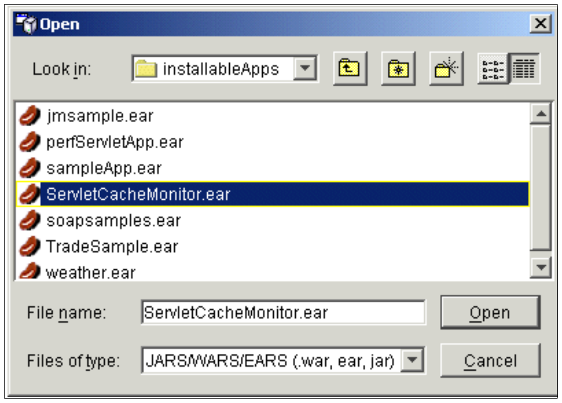
Figure 8-25 Installation complete window

10. The application has now been installed.
11. Repeat the whole procedure for the Storm customer with the associated virtual host and ServerGroup.

Install the cache monitor application

To check the contents of the local cache, we install an application that allows us to look inside the WebSphere cache. This application is provided with WebSphere Application Server and is copied to the installableApps directory during the installation process.

1. Go back to the Admin Client and right-click the **Enterprise Application** item in the left-hand pane. Select **Install Enterprise Application**.
2. Browse the installableApps directory.

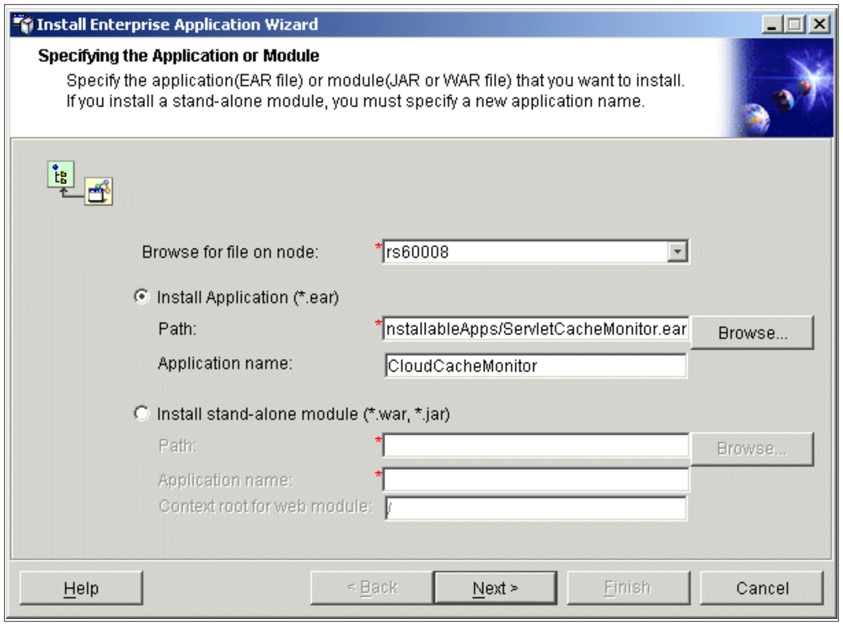


File name: ServletCacheMonitor.ear

Files of type: JARS/WARS/EARS (.war, ear, jar)

Figure 8-26   Select the Cache Monitor Application

3. Select the ServletCacheMonitor application and click **Open**.



Specifying the Application or Module

Specify the application(EAR file) or module(JAR or WAR file) that you want to install.
If you install a stand-alone module, you must specify a new application name.

Browse for file on node: *rs60008

(●) Install Application (*.ear)

Path: *nstallableApps/ServletCacheMonitor.ear

Application name: CloudCacheMonitor

( ) Install stand-alone module (*.war, *.jar)

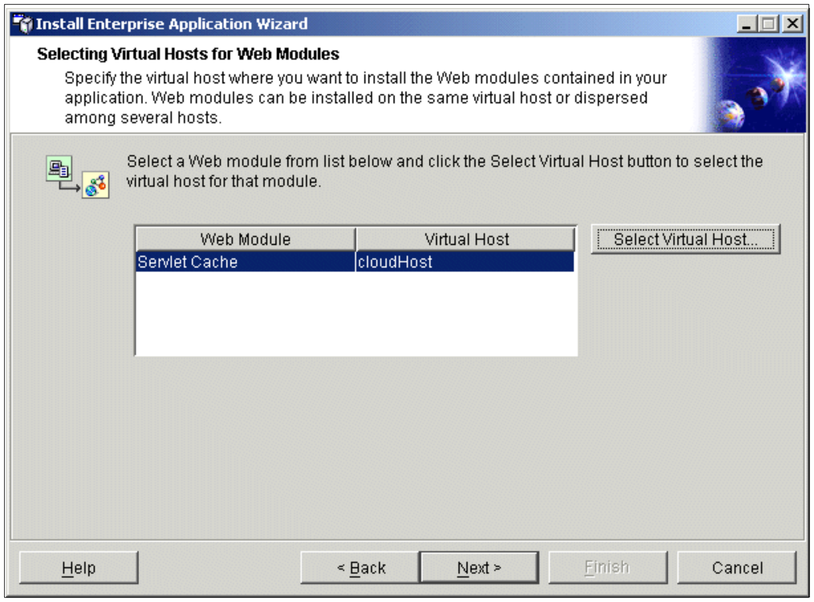
Path: *

Application name: *

Context root for web module: /

Figure 8-27   Specify a name for the application

4. Specify a name for the application. We chose CloudCacheMonitor.


Install Enterprise Application Wizard

Selecting Virtual Hosts for Web Modules

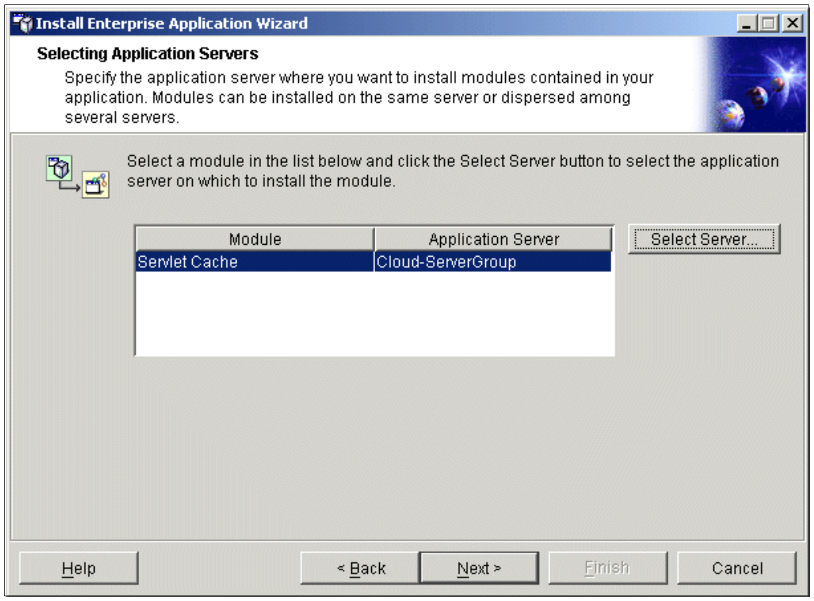
Specify the virtual host where you want to install the Web modules contained in your application. Web modules can be installed on the same virtual host or dispersed among several hosts.

Select a Web module from list below and click the Select Virtual Host button to select the virtual host for that module.

Web Module	Virtual Host
Servlet Cache	cloudHost

Figure 8-28 Select the virtual host

5. Click **Next** in all windows until you reach the window to select the virtual host for the application. Select the **CloudHost** again. Click **Next** to proceed.


Install Enterprise Application Wizard

Selecting Application Servers

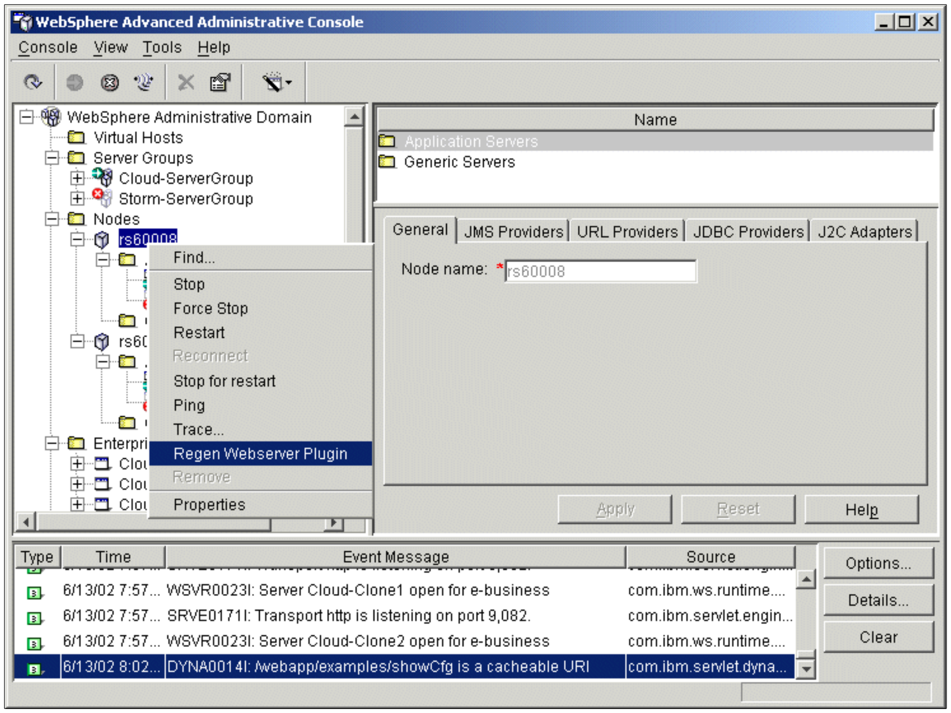
Specify the application server where you want to install modules contained in your application. Modules can be installed on the same server or dispersed among several servers.

Select a module in the list below and click the Select Server button to select the application server on which to install the module.

Module	Application Server
Servlet Cache	Cloud-ServerGroup

Figure 8-29 Select the application server for the application

6. In the next window, select **Cloud-ServerGroup** to run the application on. From here on out, it is the same process as before. You do not to regenerate the code.
7. Repeat the same procedure for the Storm company with stormHost and Storm-ServerGroup.
8. When you have finished all of this, the installedApps directory will contain four new directories:
   – CloudWeatherApp.ear
   – CloudCacheMonitor.ear
   – StormWeatherApp.ear
   – StormCacheMonitor.ear
9. You must copy the directories to the second application server node. Use the tar or zip program to pack both directories into an archive and send it via FTP to the other node. Don't forget to switch to binary mode to send an archive file via FTP.
10. On the target server, unpack the archive with the proper command.



Node name: * rs60008

Type	Time	Event Message	Source
	6/13/02 7:57...	WSVR0023I: Server Cloud-Clone1 open for e-business	com.ibm.ws.runtime....
	6/13/02 7:57...	SRVE0171I: Transport http is listening on port 9,082.	com.ibm.servlet.engin...
	6/13/02 7:57...	WSVR0023I: Server Cloud-Clone2 open for e-business	com.ibm.ws.runtime....
	6/13/02 8:02...	DYNA0014I: /webapp/examples/showCfg is a cacheable URI	com.ibm.servlet.dyna...

Figure 8-30 Plug-in configuration regeneration

11. Before you can connect to the application, you have to regenerate the plugin-cfg.xml file. Right-click each node and select **Regen Webserver Plugin**.
12. Restart the HTTP Servers on each node.
13. Your applications are now ready.

8.2.5 HTTP server

Virtual hosts

As long as all content is delivered by WebSphere Application Server through the file servlet, you will not need to define any virtual hosts in the HTTP server's httpd.conf configuration file. All that needs to be done is to add a LISTEN directive for port 81. The default path on AIX for the file is:

```
/usr/HTTPServer/conf/httpd.conf
```

Add the directive:

```
Listen 81
```

Alias the loopback adapter

Because we use the MAC forwarding method for load balancing, we must add the content cluster address as an alias on both Web Application Servers' loopback adapter. For details on this task see 6.2.1, "Aliasing a loopback adapter" on page 165. The cluster address we used is:

```
192.168.10.70
```

8.2.6 Caching proxies

Alias the loopback adapter

Because we use the MAC forwarding method for load balancing, we must add the proxy cluster address as an alias on both caching proxy nodes' loopback adapter. For details on this task, see 6.2.1, "Aliasing a loopback adapter" on page 165. The cluster address we used is:

```
192.168.10.60
```

Proxy server configuration

We configured the caching proxy by editing the ibmproxy.conf file manually. We edited or verified the following entries.

Basic configuration

- `ServerRoot C:\Progra~1\IBM\edge\cp\bin`
- `HostName m23bk63z.itso.ral.ibm.com`
- `Port 80`
- `AdminPort 8008`

Mapping rules

To enable proxying to the content servers, we need to add mapping rules:

- `proxy /* http://cloudhost:81/* www.cloud`
- `proxy /* http://stormhost:81/* www.storm`

Cache behavior

The directives to configure the caching proxy for memory caching:

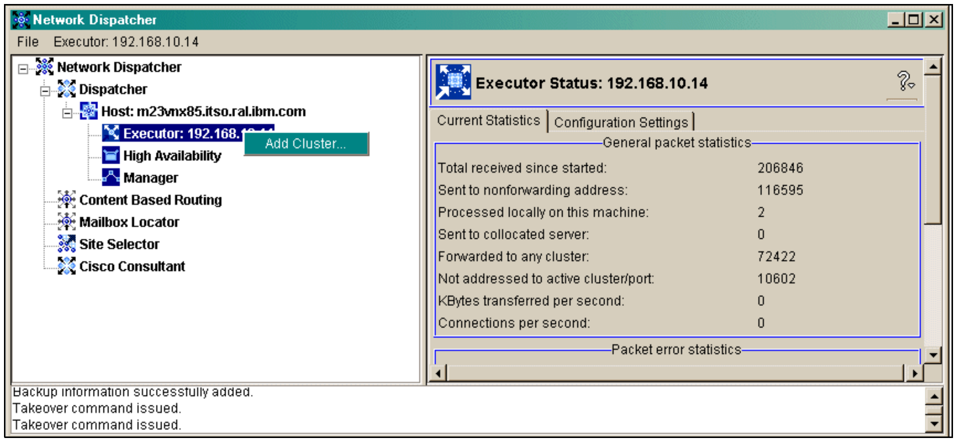
- `Cache ON`
- `#CacheDev \\.\D: (Make sure this is commented out for memory caching)`
- `CacheMemory 64 M`

8.2.7 Load balancers

The load balancer nodes will have two clusters configured. The first cluster balances the request between the caching proxy nodes. The second cluster receives the forwarded requests from the caching proxies and balances them to the content servers. Refer to Figure 8-2 on page 229 and Figure 8-3 on page 230 for details.

Create the caching proxy cluster

1. Start the Network Dispatcher admin GUI and connect to the host.

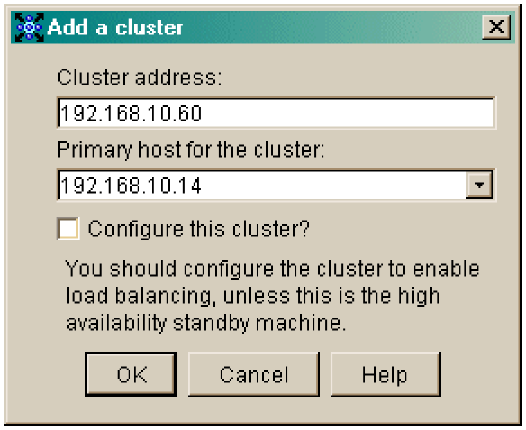


Total received since started:	206846
Sent to nonforwarding address:	116595
Processed locally on this machine:	2
Sent to collocated server:	0
Forwarded to any cluster:	72422
Not addressed to active cluster/port:	10602
KBytes transferred per second:	0
Connections per second:	0

```
Backup information successfully added.
Takeover command issued.
Takeover command issued.
```

Figure 8-31 Add a cluster

2. Right-click the **Executor** item in the left-hand pane and select **AddCluster**.



Cluster address: 192.168.10.60

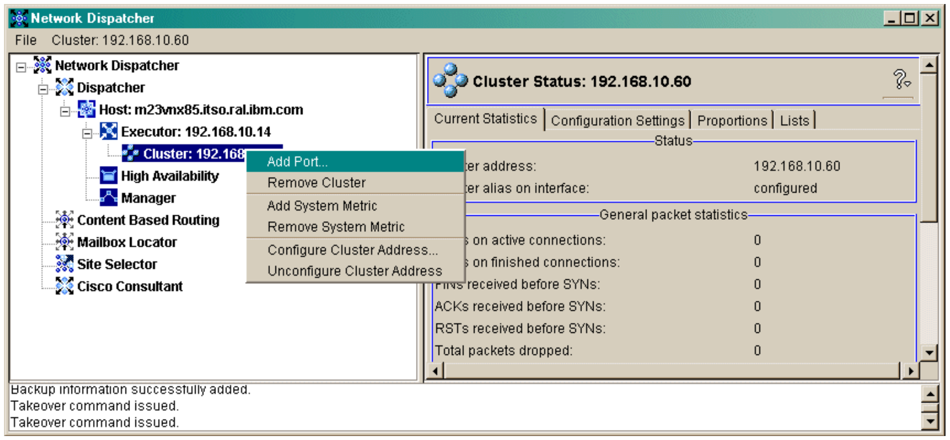
Primary host for the cluster: 192.168.10.14

☐ Configure this cluster?

You should configure the cluster to enable load balancing, unless this is the high availability standby machine.

Figure 8-32 Specify the cluster address

3. Fill in the Cluster address field. The load balancer host that you are connected to will be shown as the primary host for the cluster. Do not select the **Configure this cluster?** checkbox, as we will configure the load balancers in high availability mode.



Status

ter address:	192.168.10.60
ter alias on interface:	configured

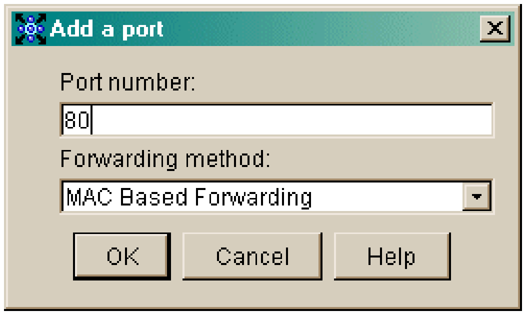
General packet statistics

s on active connections:	0
s on finished connections:	0
FINs received before SYNs:	0
ACKs received before SYNs:	0
RSTs received before SYNs:	0
Total packets dropped:	0

```
Backup information successfully added.
Takeover command issued.
Takeover command issued.
```

Figure 8-33 Add port

4. Once the cluster is added, right-click its item in the left-hand pane and select **Add Port**.

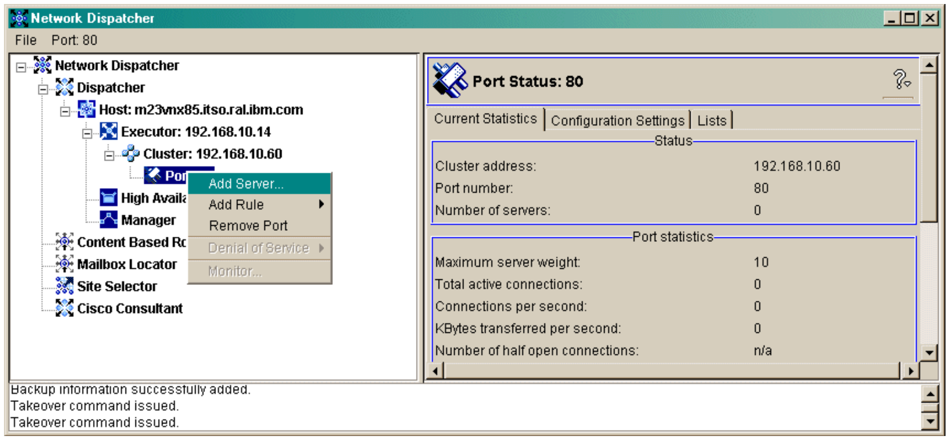


Port number: 80

Forwarding method: MAC Based Forwarding

Figure 8-34 Define the port number

5. Define the port number to be added to the cluster. We chose port 80, as this will be the port accessed from the Internet, as shown in Figure 8-3 on page 230. The proxy servers are within the same subnet, so we accept the MAC forwarding method.

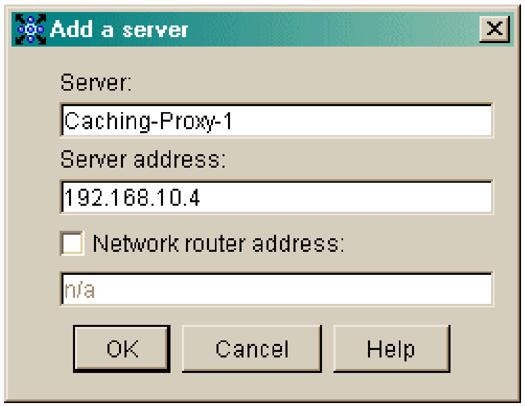


Cluster address:	192.168.10.60
Port number:	80
Number of servers:	0
Maximum server weight:	10
Total active connections:	0
Connections per second:	0
KBytes transferred per second:	0
Number of half open connections:	n/a

```
Backup information successfully added.
Takeover command issued.
Takeover command issued.
```

Figure 8-35 Add Server

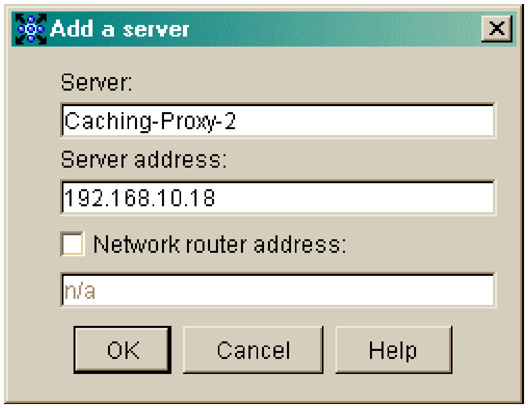
6. When the port is added, right-click the **Port** item in the left-hand pane and select **Add Server**.



Server: Caching-Proxy-1
Server address: 192.168.10.4
Network router address: n/a

Figure 8-36 Specify server name and address

7. Specify a name and the IP address for the server to be added. The name can be a display name or the real host name. If you specify the real hostname, it will find the IP address itself. Otherwise, fill in the IP address. We chose to use more descriptive names for our servers. Do not fill in a network router address. This is only used if a remote load balancer is used in a WAN load balancing topology.



Server:
Caching-Proxy-2
Server address:
192.168.10.18
☐ Network router address:
n/a

Figure 8-37 Adding the second server

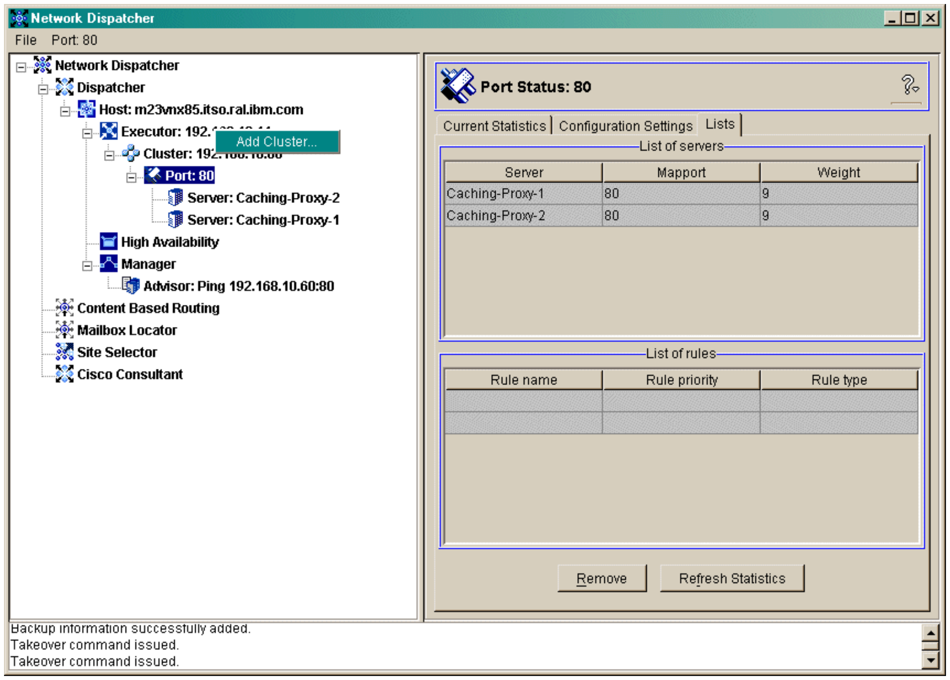
8. Repeat this procedure for all caching proxy servers that will join this cluster.

Enabling SSL connections

If you want to enable SSL connections, add a second port (443) to this cluster and add the same servers to it. Your caching proxy servers would then need to be configured for SSL. Load Balancer will then tunnel the request to the caching proxy servers without decrypting it.

Create the Content Server cluster

In order to balance the content servers, we add a second load balanced cluster.



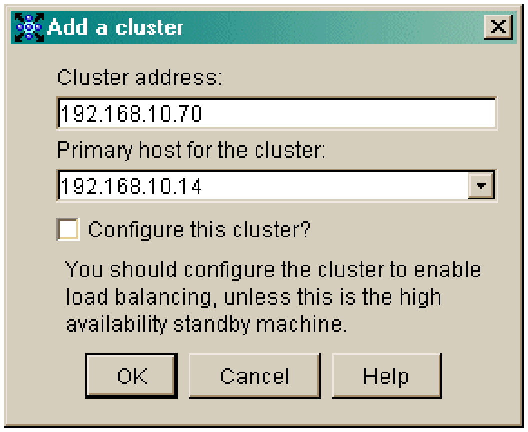
Server	Mapport	Weight
Caching-Proxy-1	80	9
Caching-Proxy-2	80	9

Rule name	Rule priority	Rule type

```
Backup information successfully added.
Takeover command issued.
Takeover command issued.
```

Figure 8-38   Adding the second cluster

1. Again, right-click the **Executor** item and select **Add Cluster**.



Cluster address: 192.168.10.70

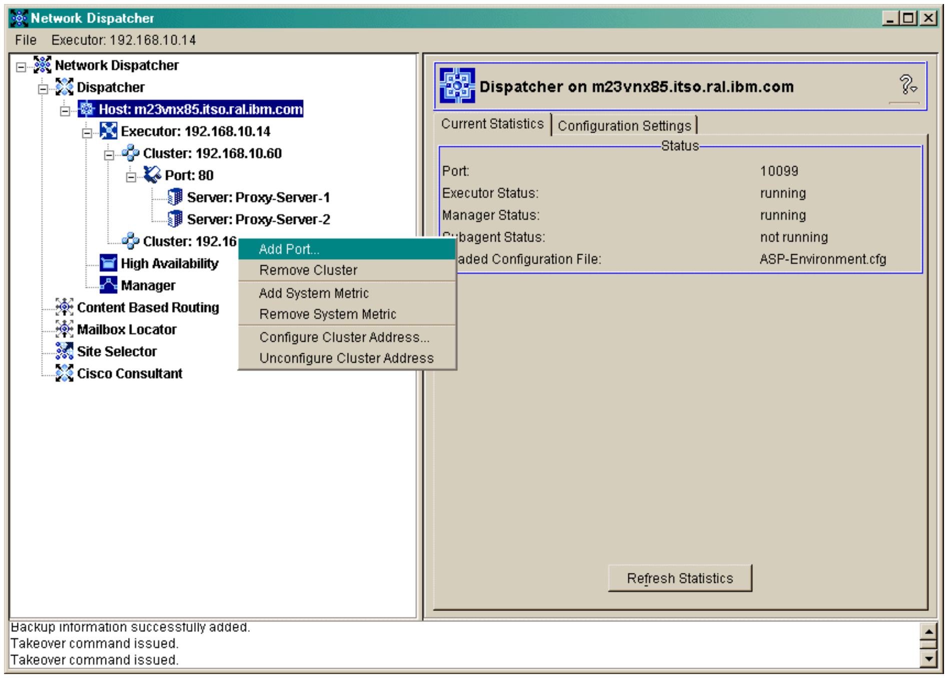
Primary host for the cluster: 192.168.10.14

☐ Configure this cluster?

You should configure the cluster to enable load balancing, unless this is the high availability standby machine.

Figure 8-39   Specify the cluster address

2. Specify the cluster's IP address and the primary host for it. Also, do not select the **Configure this cluster?** check box.

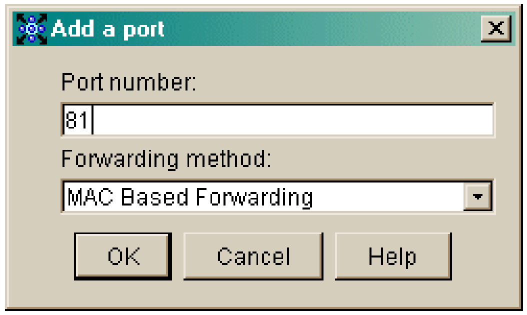


Port: 10099
Executor Status: running
Manager Status: running
Subagent Status: not running
Loaded Configuration File: ASP-Environment.cfg

```
Backup information successfully added.
Takeover command issued.
Takeover command issued.
```

Figure 8-40 Adding a port to the second cluster

3. Right-click the second cluster item and select **Add Port**.

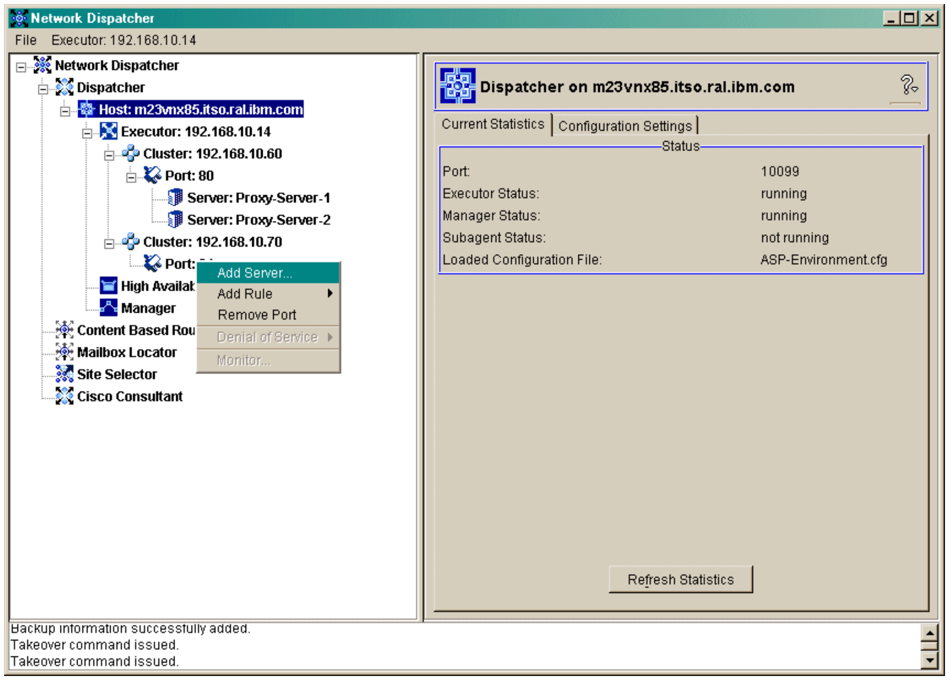


Port number: 81

Forwarding method: MAC Based Forwarding

Figure 8-41 Specify the port number for the second cluster

4. This time, we do not want the cluster to be accessible from the Internet. For this reason, we chose port 81. Again, we chose the MAC forwarding method.

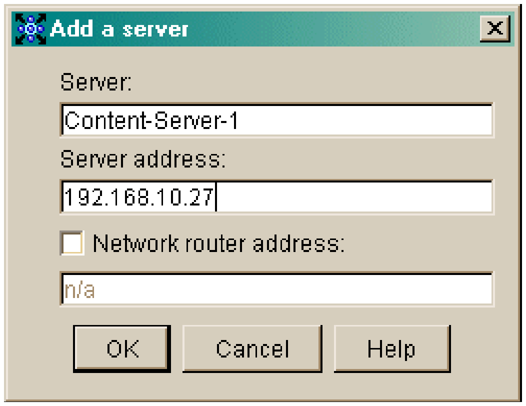


Port:	10099
Executor Status:	running
Manager Status:	running
Subagent Status:	not running
Loaded Configuration File:	ASP-Environment.cfg

```
Backup information successfully added.
Takeover command issued.
Takeover command issued.
```

Figure 8-42 Adding a server to the second cluster

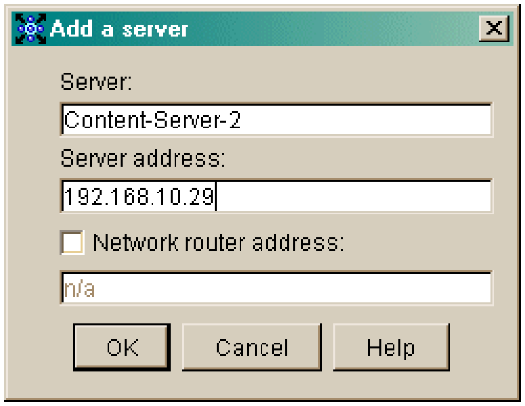
5. Right-click the **Port** item of the second cluster to add a server.



Server:
Content-Server-1
Server address:
192.168.10.27
☐ Network router address:
n/a

Figure 8-43 Adding the first server to port 81

6. This time, we added the WebSphere Application servers to the port.



Server:
Content-Server-2
Server address:
192.168.10.29
☐ Network router address:
n/a

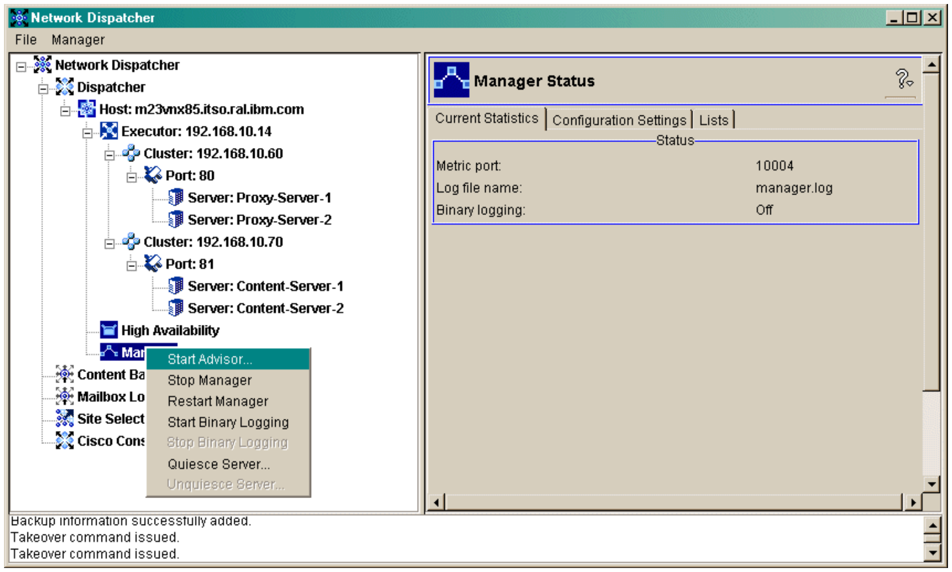
Figure 8-44 Adding the second server

7. Repeat this procedure for all content servers that will join this cluster.

Adding advisors to the clusters

Advisors enable the load balancer manager to monitor the servers in a cluster. The response time to answer the request is the base for a number that is being calculated. This number is known as the *load*. It will determine what percentage of the overall load on a cluster is distributed to each server. Refer to 4.2.8, “Advisors” on page 76 for more information.

1. If it is not running, start the manager from the executor’s context menu.

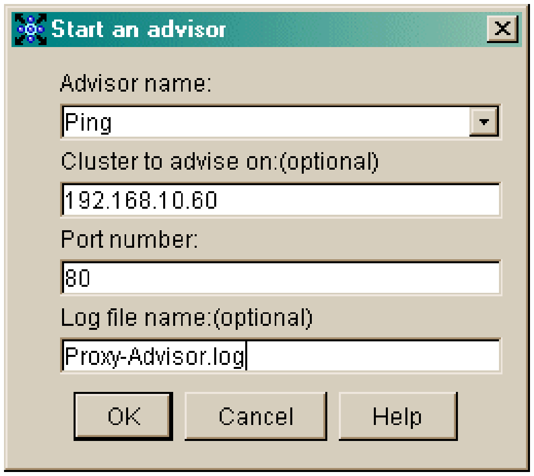


Metric port:	10004
Log file name:	manager.log
Binary logging:	Off

```
Backup information successfully added.
Takeover command issued.
Takeover command issued.
```

Figure 8-45 Adding an advisor to a cluster

2. Right-click the manager and select **Start Advisor**.

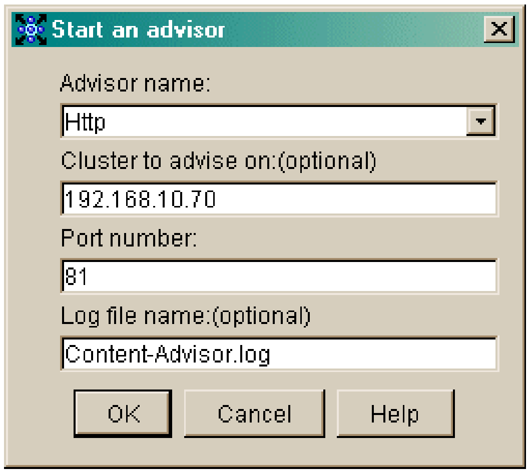


Advisor name: Ping
Cluster to advise on:(optional) 192.168.10.60
Port number: 80
Log file name:(optional) Proxy-Advisor.log

Figure 8-46 Specify an advisor

3. Select an advisor type from the advisor name drop-down list. Unfortunately, the caching proxy advisor did not receive a valid response when the caching proxy was running on port 80. On a different port, it works normally. So we chose the HTTP advisor instead. Also specify the cluster to advise on, the port number and, optionally, a name for the log file. A descriptive log file name

is quite helpful. If you have ever taken a look at the logs directory, you are familiar with the concept.



Advisor name: Http

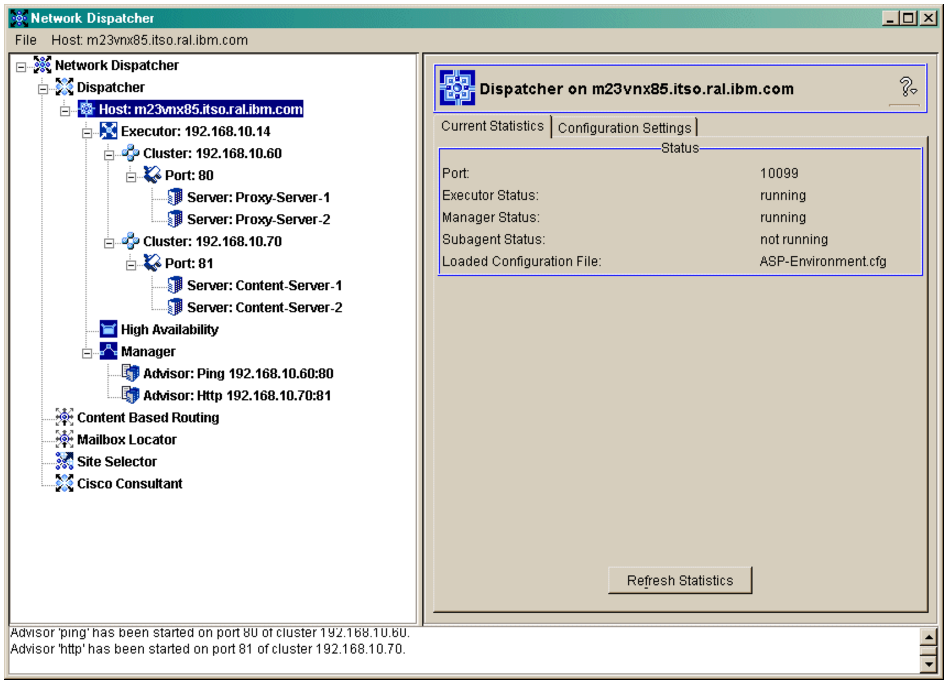
Cluster to advise on:(optional) 192.168.10.70

Port number: 81

Log file name:(optional) Content-Advisor.log

Figure 8-47 Starting the HTTP advisor

4. We repeated the procedure to add another advisor on our WebSphere Application server cluster.



Port:	10099
Executor Status:	running
Manager Status:	running
Subagent Status:	not running
Loaded Configuration File:	ASP-Environment.cfg

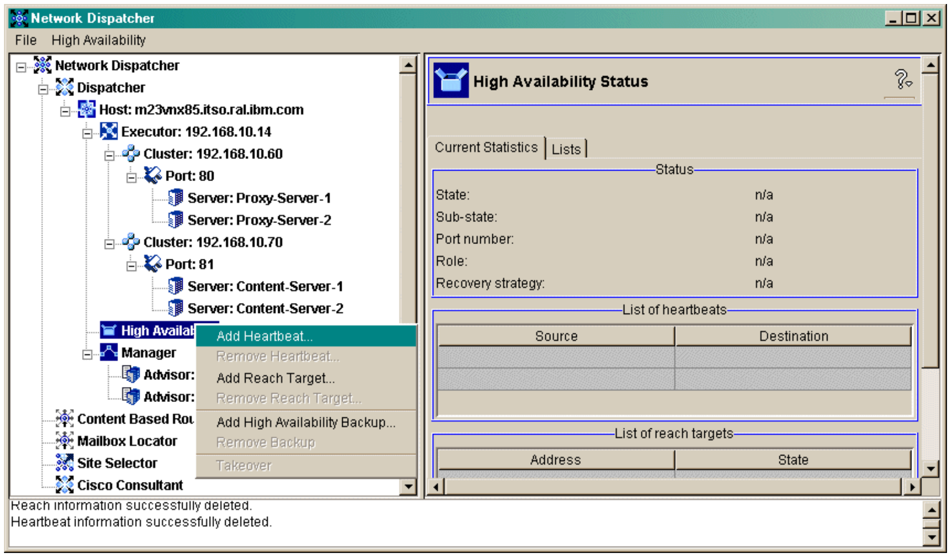
```
Advisor 'ping' has been started on port 80 of cluster 192.168.10.60.
Advisor 'http' has been started on port 81 of cluster 192.168.10.70.
```

Figure 8-48 Overview

5. Figure 12-29 shows an overview of what we have configured so far.

Configure high availability

The next step is to configure high availability for the load balancer node.

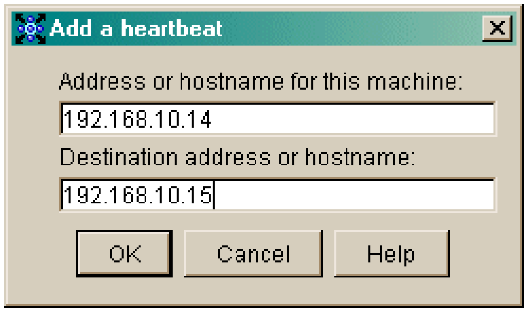


State: n/a
Sub-state: n/a
Port number: n/a
Role: n/a
Recovery strategy: n/a

Reach information successfully deleted.
Heartbeat information successfully deleted.

Figure 8-49 Adding the heartbeat

1. Right-click **High availability** and select **Add Heartbeat**.

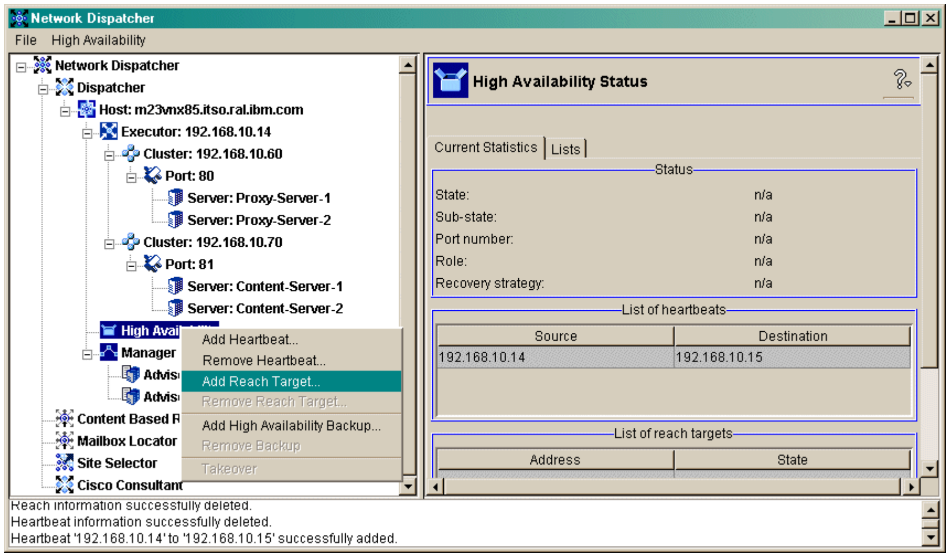


Address or hostname for this machine: 192.168.10.14

Destination address or hostname: 192.168.10.15

Figure 8-50 Heartbeat window

2. The backup load balancer is the destination address to add for the heartbeat.



State:	n/a
Sub-state:	n/a
Port number:	n/a
Role:	n/a
Recovery strategy:	n/a

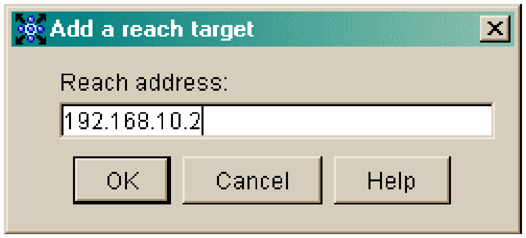
Source	Destination
192.168.10.14	192.168.10.15

Address	State

```
Reach information successfully deleted.
Heartbeat information successfully deleted.
Heartbeat '192.168.10.14' to '192.168.10.15' successfully added.
```

Figure 8-51 Adding a reach target

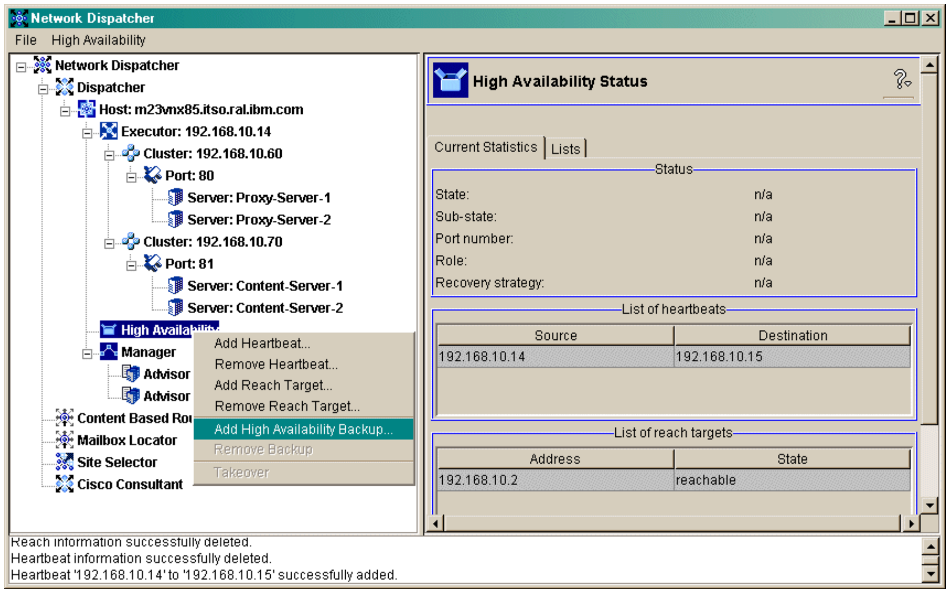
3. Right-click **High Availability** and select **Add Reach Target**.



Reach address: 192.168.10.2

Figure 8-52 Specify a target address

4. A network router is good to use as the reach target. Fill in the IP address of a target. It helps the load balancer to determine whether it has network access. In the case of the heartbeat stopping, this helps to determine whether to take over or not.



Status	
State:	n/a
Sub-state:	n/a
Port number:	n/a
Role:	n/a
Recovery strategy:	n/a

List of heartbeats

Source	Destination
192.168.10.14	192.168.10.15

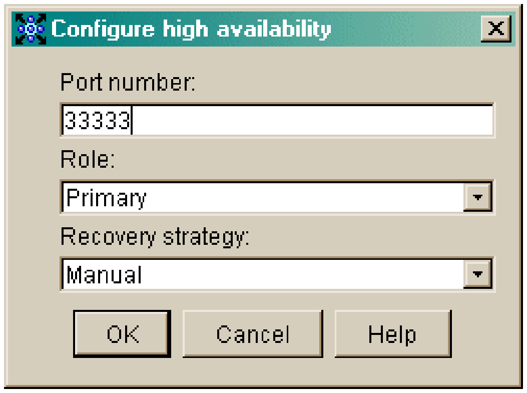
List of reach targets

Address	State
192.168.10.2	reachable

```
Reach information successfully deleted.
Heartbeat information successfully deleted.
Heartbeat '192.168.10.14' to '192.168.10.15' successfully added.
```

Figure 8-53 Add high availability backup

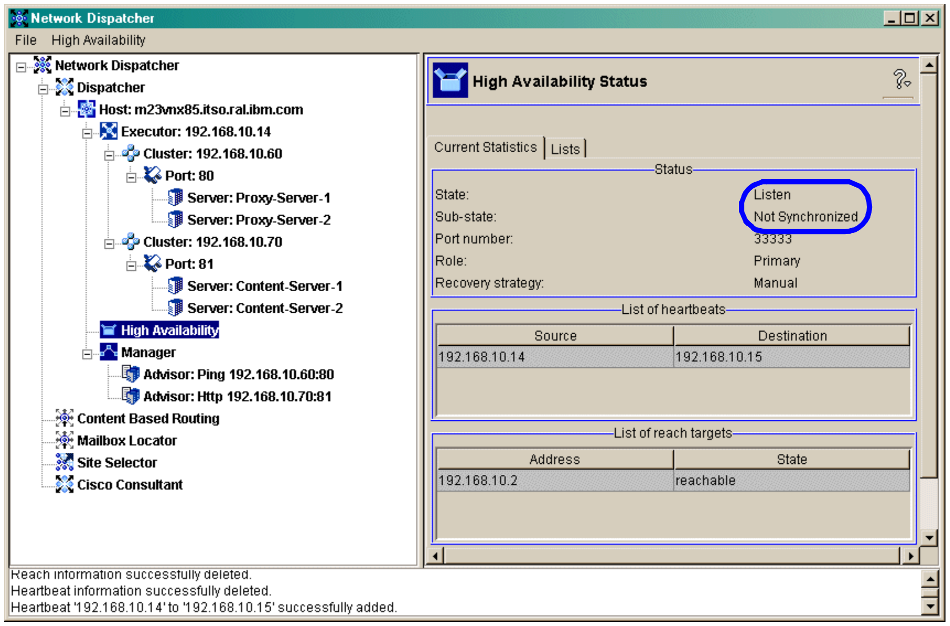
5. Right-click **High Availability** and select **Add High Availability Backup**.



Port number: 33333
Role: Primary
Recovery strategy: Manual

Figure 8-54 HA role definition

6. Specify a port number for the heartbeat. It can be any number smaller than 65536. Select the primary role for this load balancer node. If you want to be able to issue takeover commands, chose the manual recovery strategy.



Status

State:	Listen
Sub-state:	Not Synchronized
Port number:	33333
Role:	Primary
Recovery strategy:	Manual

List of heartbeats

Source	Destination
192.168.10.14	192.168.10.15

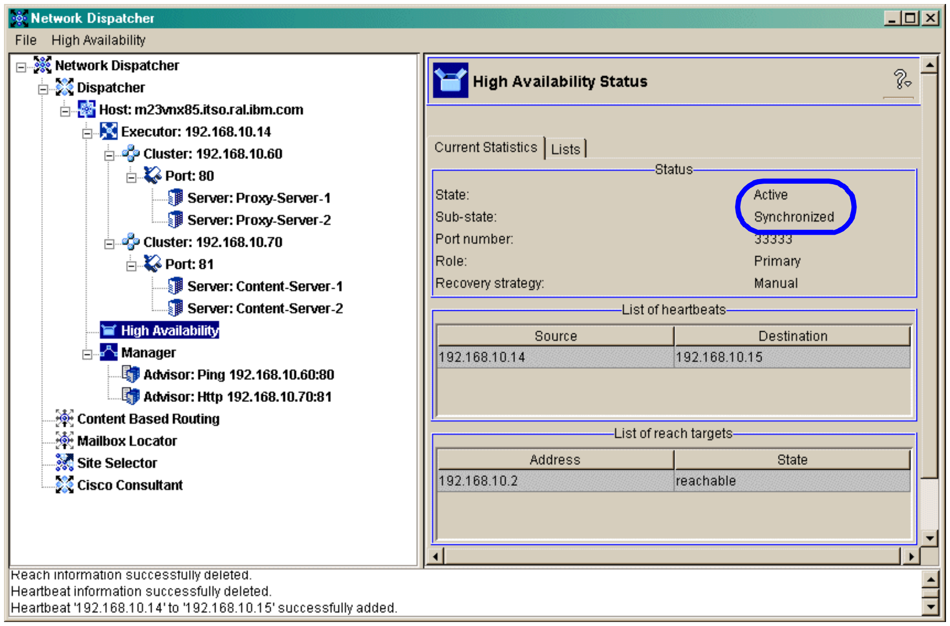
List of reach targets

Address	State
192.168.10.2	reachable

```
Reach information successfully deleted.
Heartbeat information successfully deleted.
Heartbeat '192.168.10.14' to '192.168.10.15' successfully added.
```

Figure 8-55 HA configured

7. Figure 12-36 shows the configured high availability. Note that the state is Listen and the sub-state is Not synchronized. This is because the backup load balancer does not yet respond to the heartbeats.
8. Repeat the high availability steps on the backup load balancer node.



Status

State:	Active
Sub-state:	Synchronized
Port number:	33333
Role:	Primary
Recovery strategy:	Manual

List of heartbeats

Source	Destination
192.168.10.14	192.168.10.15

List of reach targets

Address	State
192.168.10.2	reachable

```
Reach information successfully deleted.
Heartbeat information successfully deleted.
Heartbeat '192.168.10.14' to '192.168.10.15' successfully added.
```

Figure 8-56 High availability now synchronized

9. Figure 12-37 shows that after configuring the backup load balancer node, they now are synchronized. This means that they are exchanging heartbeats and with that, synchronizes backup's connection table.

Edit the takeover scripts

The load balancers have the cluster's IP addresses aliased to the network interfaces. The primary load balancer has the cluster addresses aliased to the physical token ring or ethernet interface, while the backup load balancer has the cluster addresses aliased to the loopback adapter. To take over from one machine to the other, these aliases must be changed. For this purpose, we need to prepare takeover scripts. They are located in the load balancer's binary directory. There are four different scripts:

- goActive

  Aliases the cluster addresses to the token ring or ethernet adapter and deletes the ones on the loopback.

- goStandby

  Aliases the cluster addresses to the loopback adapter and deletes the ones on the token ring or ethernet adapter.

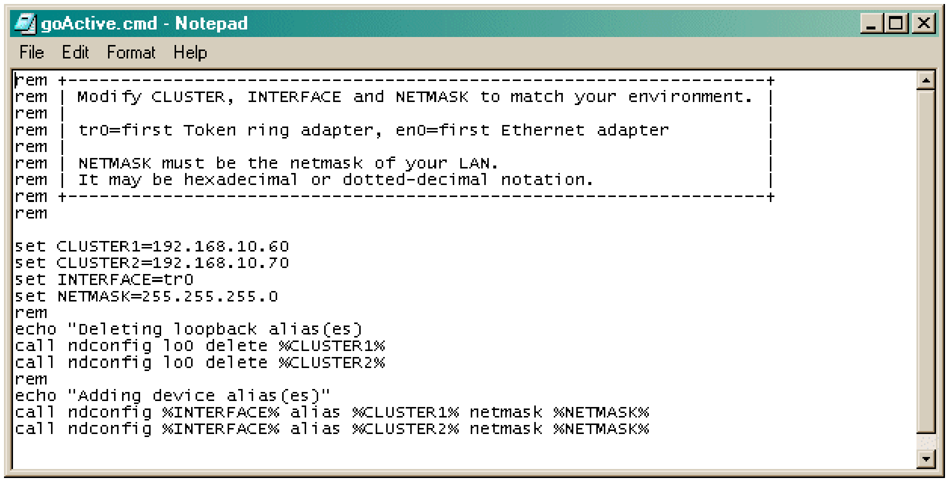
- goIdle

  Aliases the cluster addresses to the token ring or ethernet adapter only.

- goInOp

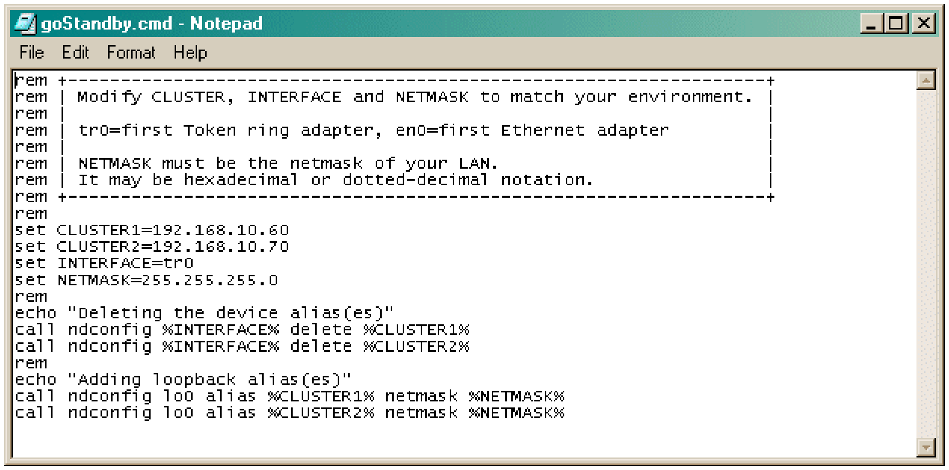
  Deletes the cluster addresses from both: the token ring or ethernet adapter and the loopback adapter.

The following screen shots show you the edited scripts prepared for our environment.



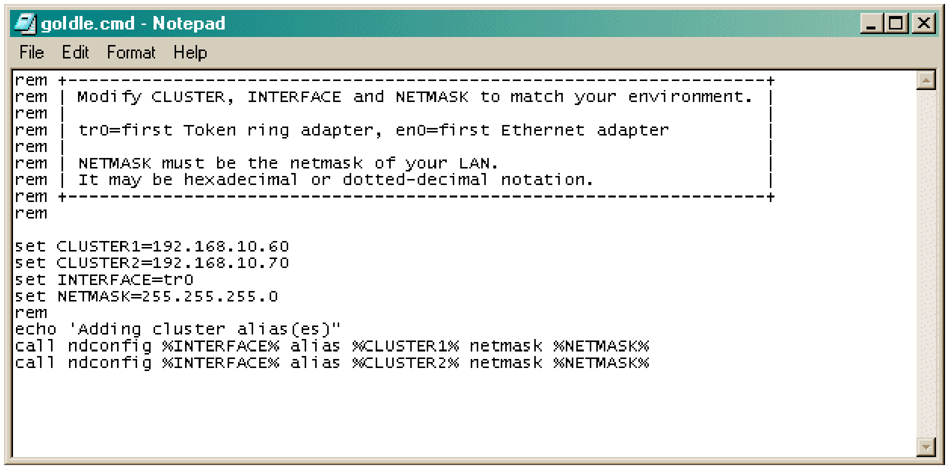
```
rem +---------------------------------------------------------------------+
rem | Modify CLUSTER, INTERFACE and NETMASK to match your environment. |
rem |                                                                     |
rem | tr0=first Token ring adapter, en0=first Ethernet adapter           |
rem |                                                                     |
rem | NETMASK must be the netmask of your LAN.                           |
rem | It may be hexadecimal or dotted-decimal notation.                  |
rem +---------------------------------------------------------------------+
rem

set CLUSTER1=192.168.10.60
set CLUSTER2=192.168.10.70
set INTERFACE=tr0
set NETMASK=255.255.255.0
rem
echo "Deleting loopback alias(es)
call ndconfig lo0 delete %CLUSTER1%
call ndconfig lo0 delete %CLUSTER2%
rem
echo "Adding device alias(es)"
call ndconfig %INTERFACE% alias %CLUSTER1% netmask %NETMASK%
call ndconfig %INTERFACE% alias %CLUSTER2% netmask %NETMASK%
```

Figure 8-57 goActive.cmd



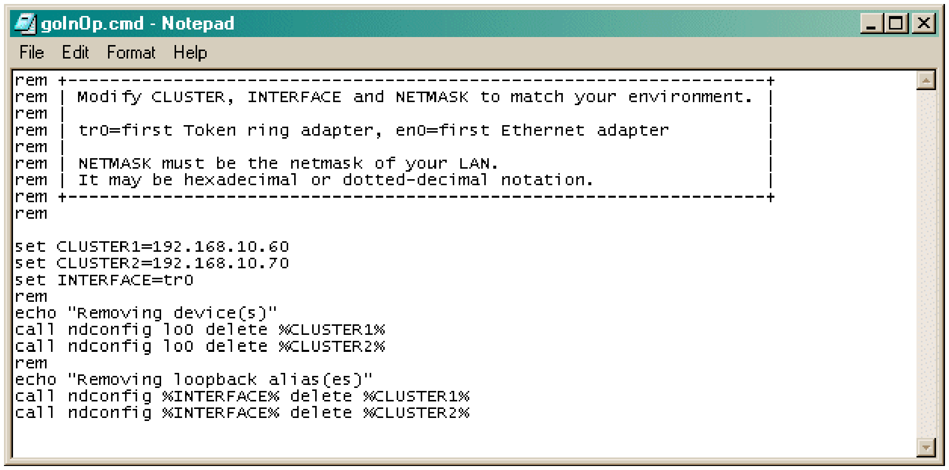
```
rem +---------------------------------------------------------------------+
rem | Modify CLUSTER, INTERFACE and NETMASK to match your environment. |
rem |                                                                     |
rem | tr0=first Token ring adapter, en0=first Ethernet adapter           |
rem |                                                                     |
rem | NETMASK must be the netmask of your LAN.                           |
rem | It may be hexadecimal or dotted-decimal notation.                  |
rem +---------------------------------------------------------------------+
rem
set CLUSTER1=192.168.10.60
set CLUSTER2=192.168.10.70
set INTERFACE=tr0
set NETMASK=255.255.255.0
rem
echo "Deleting the device alias(es)"
call ndconfig %INTERFACE% delete %CLUSTER1%
call ndconfig %INTERFACE% delete %CLUSTER2%
rem
echo "Adding loopback alias(es)"
call ndconfig lo0 alias %CLUSTER1% netmask %NETMASK%
call ndconfig lo0 alias %CLUSTER2% netmask %NETMASK%
```

Figure 8-58 goStandby.cmd



```
rem +------------------------------------------------------------------+
rem | Modify CLUSTER, INTERFACE and NETMASK to match your environment. |
rem |                                                                  |
rem | tr0=first Token ring adapter, en0=first Ethernet adapter         |
rem |                                                                  |
rem | NETMASK must be the netmask of your LAN.                         |
rem | It may be hexadecimal or dotted-decimal notation.                |
rem +------------------------------------------------------------------+
rem

set CLUSTER1=192.168.10.60
set CLUSTER2=192.168.10.70
set INTERFACE=tr0
set NETMASK=255.255.255.0
rem
echo 'Adding cluster alias(es)"
call ndconfig %INTERFACE% alias %CLUSTER1% netmask %NETMASK%
call ndconfig %INTERFACE% alias %CLUSTER2% netmask %NETMASK%
```

Figure 8-59 goIdle.cmd

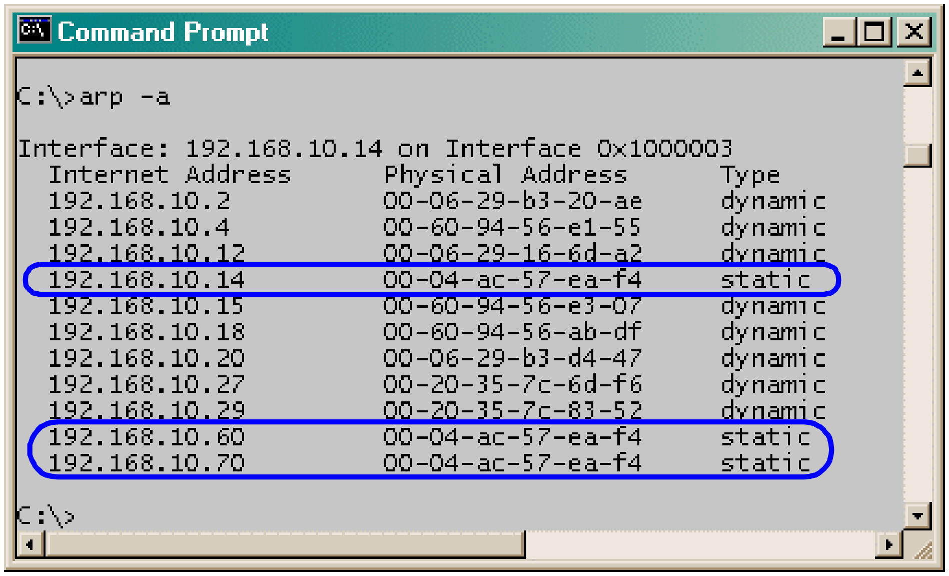


```
rem +------------------------------------------------------------------+
rem | Modify CLUSTER, INTERFACE and NETMASK to match your environment. |
rem |                                                                  |
rem | tr0=first Token ring adapter, en0=first Ethernet adapter         |
rem |                                                                  |
rem | NETMASK must be the netmask of your LAN.                         |
rem | It may be hexadecimal or dotted-decimal notation.                |
rem +------------------------------------------------------------------+
rem

set CLUSTER1=192.168.10.60
set CLUSTER2=192.168.10.70
set INTERFACE=tr0
rem
echo "Removing device(s)"
call ndconfig lo0 delete %CLUSTER1%
call ndconfig lo0 delete %CLUSTER2%
rem
echo "Removing loopback alias(es)"
call ndconfig %INTERFACE% delete %CLUSTER1%
call ndconfig %INTERFACE% delete %CLUSTER2%
```

Figure 8-60 goInOp.cmd

Test high availability

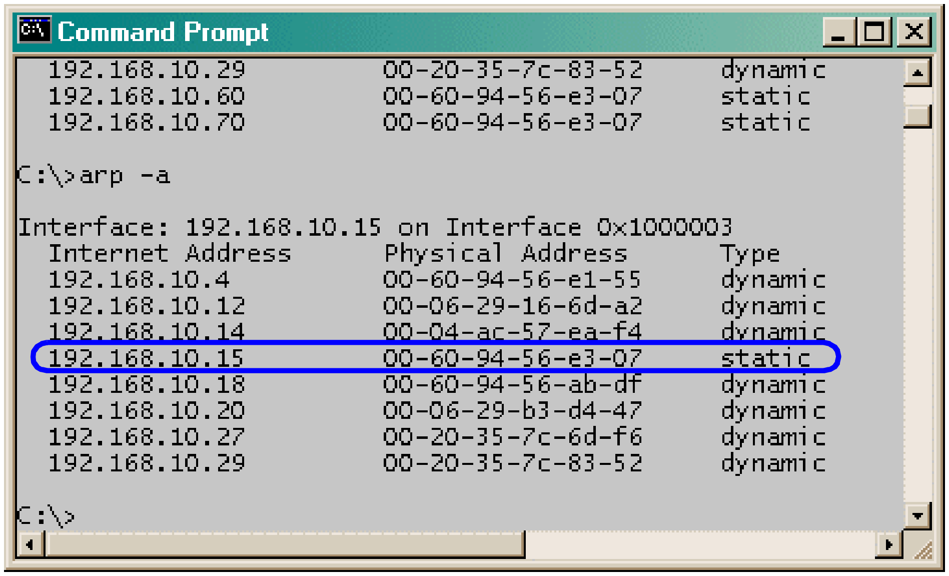
The following procedure shows how to verify that the takeover works correctly. On Windows 2000, the **`ipconfig`** command will not show the aliases because on Windows 2000, they are not actually added to the interface. Instead, a static entry in the ARP (Address Resolution Protocol) table is added. The MS Loopback Adapter does not need to be installed. On Unix platforms, a normal alias on lo0 is used.



```
C:\>arp -a

Interface: 192.168.10.14 on Interface 0x1000003
  Internet Address      Physical Address      Type
  192.168.10.2          00-06-29-b3-20-ae     dynamic
  192.168.10.4          00-60-94-56-e1-55     dynamic
  192.168.10.12         00-06-29-16-6d-a2     dynamic
  192.168.10.14         00-04-ac-57-ea-f4     static
  192.168.10.15         00-60-94-56-e3-07     dynamic
  192.168.10.18         00-60-94-56-ab-df     dynamic
  192.168.10.20         00-06-29-b3-d4-47     dynamic
  192.168.10.27         00-20-35-7c-6d-f6     dynamic
  192.168.10.29         00-20-35-7c-83-52     dynamic
  192.168.10.60         00-04-ac-57-ea-f4     static
  192.168.10.70         00-04-ac-57-ea-f4     static

C:\>
```

Figure 8-61 ARP table on the primary load balancer (192.168.10.14)

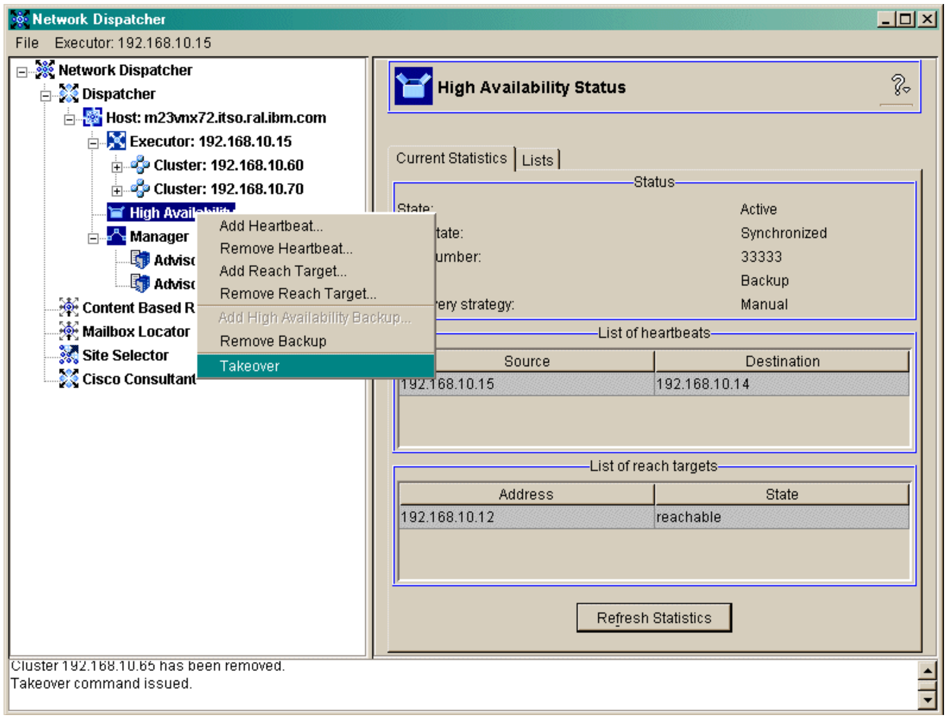
10. Open a command window on the primary load balancer machine and issue an **`arp -a`** command. The two cluster addresses are shown as static entries. Note that those and the non-forwarding address of the load balancer are the only static entries in the ARP table. See Figure 8-61 for details.



```
  192.168.10.29         00-20-35-7c-83-52     dynamic
  192.168.10.60         00-60-94-56-e3-07     static
  192.168.10.70         00-60-94-56-e3-07     static

C:\>arp -a

Interface: 192.168.10.15 on Interface 0x1000003
  Internet Address      Physical Address      Type
  192.168.10.4          00-60-94-56-e1-55     dynamic
  192.168.10.12         00-06-29-16-6d-a2     dynamic
  192.168.10.14         00-04-ac-57-ea-f4     dynamic
  192.168.10.15         00-60-94-56-e3-07     static
  192.168.10.18         00-60-94-56-ab-df     dynamic
  192.168.10.20         00-06-29-b3-d4-47     dynamic
  192.168.10.27         00-20-35-7c-6d-f6     dynamic
  192.168.10.29         00-20-35-7c-83-52     dynamic

C:\>
```

Figure 8-62 ARP table on the backup load balancer (192.168.10.15)

11. Go to the backup load balancer and go through the same procedure. This should only show the non-forwarding address as a static entry.



Status	
State:	Active
tate:	Synchronized
umber:	33333
	Backup
ery strategy:	Manual

List of heartbeats

Source	Destination
192.168.10.15	192.168.10.14

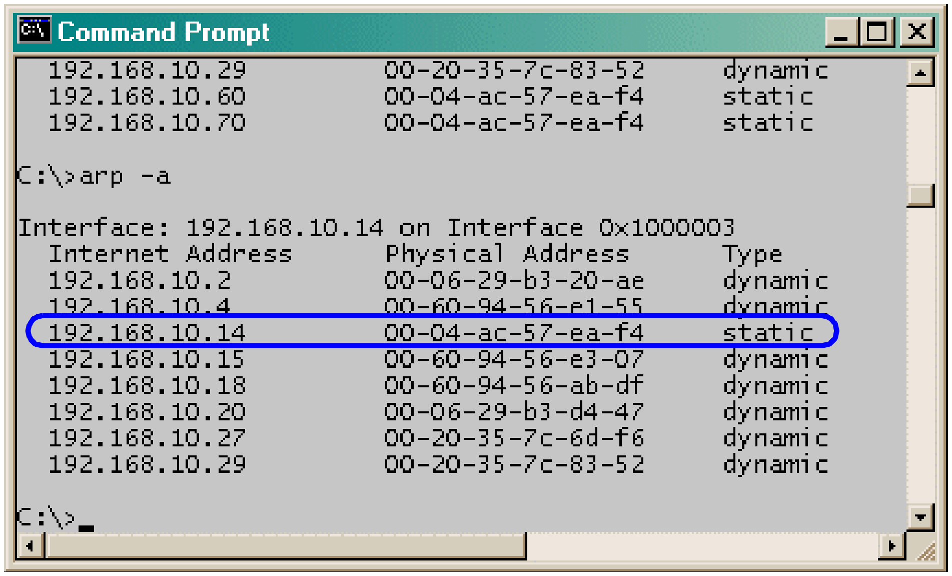
List of reach targets

Address	State
192.168.10.12	reachable

```
Cluster 192.168.10.65 has been removed.
Takeover command issued.
```

Figure 8-63 Takeover

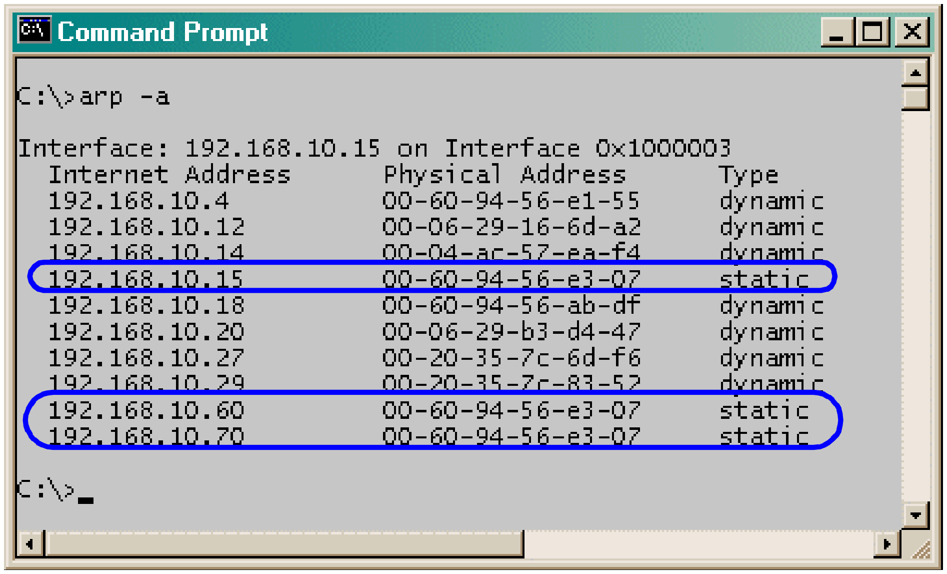
12. On the backup load balancer (192.168.10.15), right-click **High availability** and select **Takeover**.



```
  192.168.10.29         00-20-35-7c-83-52     dynamic
  192.168.10.60         00-04-ac-57-ea-f4     static
  192.168.10.70         00-04-ac-57-ea-f4     static

C:\>arp -a

Interface: 192.168.10.14 on Interface 0x1000003
  Internet Address      Physical Address      Type
  192.168.10.2          00-06-29-b3-20-ae     dynamic
  192.168.10.4          00-60-94-56-e1-55     dynamic
  192.168.10.14         00-04-ac-57-ea-f4     static
  192.168.10.15         00-60-94-56-e3-07     dynamic
  192.168.10.18         00-60-94-56-ab-df     dynamic
  192.168.10.20         00-06-29-b3-d4-47     dynamic
  192.168.10.27         00-20-35-7c-6d-f6     dynamic
  192.168.10.29         00-20-35-7c-83-52     dynamic

C:\>_
```

Figure 8-64 ARP table on former primary, now standby load balancer(192.168.10.14)

13. Back at the former primary load balancer (192.168.10.14), check the arp table again. After a moment, the cluster entries are gone (see Figure 8-64).

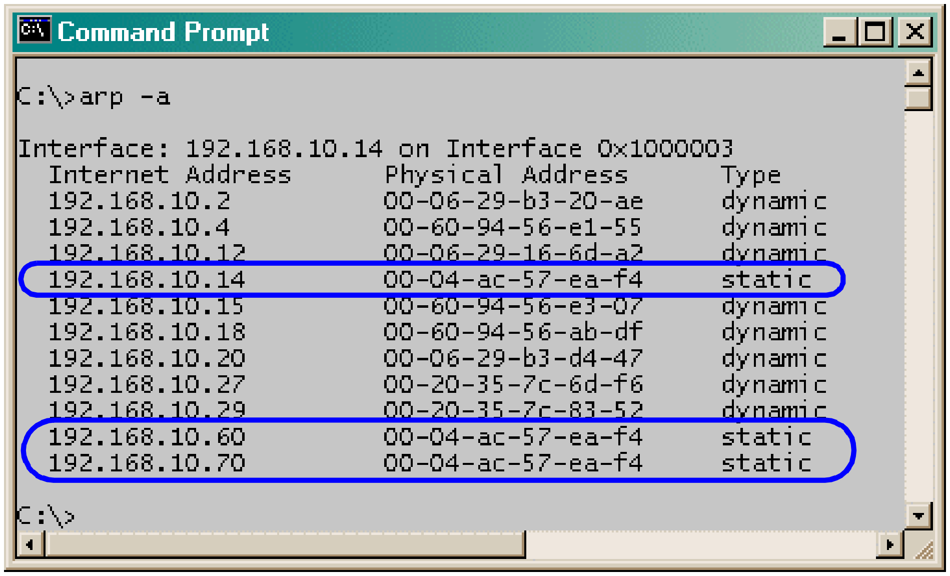


```
C:\>arp -a

Interface: 192.168.10.15 on Interface 0x1000003
  Internet Address      Physical Address      Type
  192.168.10.4          00-60-94-56-e1-55     dynamic
  192.168.10.12         00-06-29-16-6d-a2     dynamic
  192.168.10.14         00-04-ac-57-ea-f4     dynamic
  192.168.10.15         00-60-94-56-e3-07     static
  192.168.10.18         00-60-94-56-ab-df     dynamic
  192.168.10.20         00-06-29-b3-d4-47     dynamic
  192.168.10.27         00-20-35-7c-6d-f6     dynamic
  192.168.10.29         00-20-35-7c-83-52     dynamic
  192.168.10.60         00-60-94-56-e3-07     static
  192.168.10.70         00-60-94-56-e3-07     static

C:\>_
```

Figure 8-65 ARP table on former standby, now primary load balancer (192.168.10.15)

14.On the former standby, now primary load balancer (192.168.10.15), the cluster addresses show up at this point. Although this process takes a moment, no packets will be lost. During this time, they are queued and sent to the primary load balancer after the takeover process has finished.

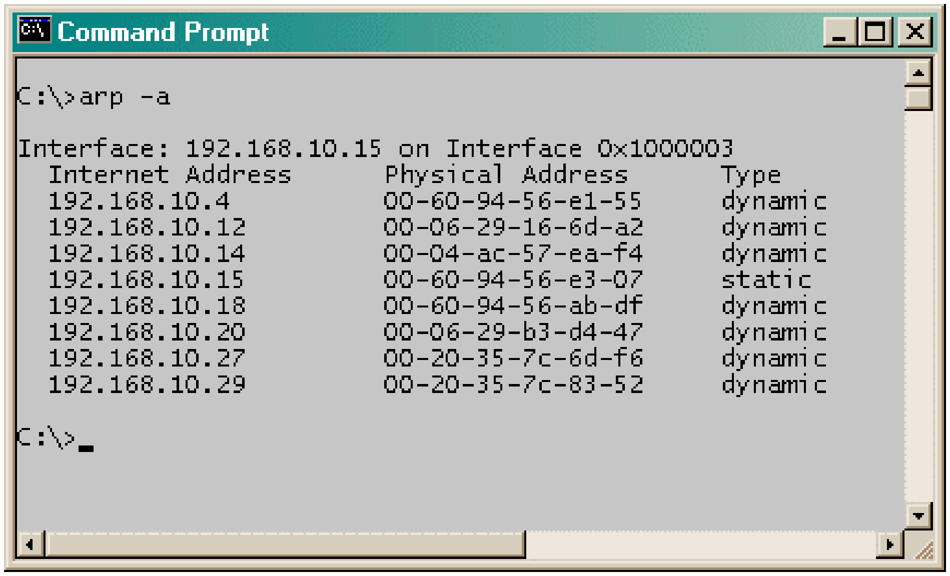
15.Both of these processes are the result of the takeover scripts being automatically invoked.



```
C:\>arp -a

Interface: 192.168.10.14 on Interface 0x1000003
  Internet Address      Physical Address      Type
  192.168.10.2          00-06-29-b3-20-ae     dynamic
  192.168.10.4          00-60-94-56-e1-55     dynamic
  192.168.10.12         00-06-29-16-6d-a2     dynamic
  192.168.10.14         00-04-ac-57-ea-f4     static
  192.168.10.15         00-60-94-56-e3-07     dynamic
  192.168.10.18         00-60-94-56-ab-df     dynamic
  192.168.10.20         00-06-29-b3-d4-47     dynamic
  192.168.10.27         00-20-35-7c-6d-f6     dynamic
  192.168.10.29         00-20-35-7c-83-52     dynamic
  192.168.10.60         00-04-ac-57-ea-f4     static
  192.168.10.70         00-04-ac-57-ea-f4     static

C:\>
```

Figure 8-66 ARP query on the again primary load balancer (192.168.10.14)

16.Now perform the takeover again from the backup machine to check that all scripts work correctly. The clusters should change back to the new primary load balancer (192.168.10.14) again. See Figure 8-66 and Figure 8-67.



```
C:\>arp -a

Interface: 192.168.10.15 on Interface 0x1000003
  Internet Address      Physical Address      Type
  192.168.10.4          00-60-94-56-e1-55     dynamic
  192.168.10.12         00-06-29-16-6d-a2     dynamic
  192.168.10.14         00-04-ac-57-ea-f4     dynamic
  192.168.10.15         00-60-94-56-e3-07     static
  192.168.10.18         00-60-94-56-ab-df     dynamic
  192.168.10.20         00-06-29-b3-d4-47     dynamic
  192.168.10.27         00-20-35-7c-6d-f6     dynamic
  192.168.10.29         00-20-35-7c-83-52     dynamic

C:\>_
```

Figure 8-67 ARP query on the again standby load balancer (192.168.10.15)

8.3 Configure WebSphere Application Server for local dynacache

For a brief review of dynamic caching, see “Dynamic caching” on page 84.

8.3.1 Web Application Servers

To configure WebSphere Application Server for caching of dynamically generated responses from servlet or JSP calls, you have to configure two elements.

1. Enable servlet caching on an application server (JVM) or a ServerGroup. Refer to Enable servlet caching below.
2. Define which servlet or JSP results are cachable. Refer to “Define cachable servlets and JSPs” on page 278.

Enable servlet caching

There are two ways to enable servlet caching.

- One way is to rename the sample properties file for the dynacache feature:

  `$WAS_ROOT/properties/dynacache.sample.xml` to

```
$WAS_ROOT/properties/dynacache.xml
```

The dynacache.xml file contains the definition of the servlet cache. Without the comments, the file looks like Example 8-2:

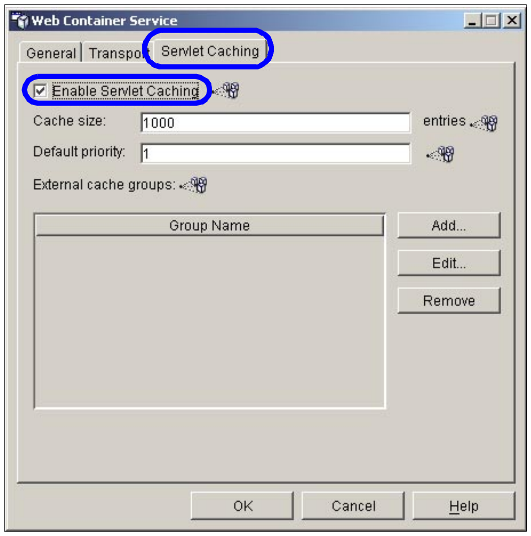
Example 8-2 dynacache.xml

```
<?xml version="1.0"?>
<!DOCTYPE cacheUnit SYSTEM "dynacache.dtd">

<cacheUnit>
  <cache size="1000" priority="1" />
</cacheUnit>
```

It defines the possible number of cache entries.

- The other way to enable servlet caching is via the GUI.

1. Go to the WebSphere Application Server Admin Console.
2. In the left-hand pane, select an Application Server or a ServerGroup.
3. Then, in the right-hand pane, select the **Services** tab. A list of services appears within this pane.
4. Select **Web Container Service** and click **Edit Properties**.



[✓] Enable Servlet Caching
Cache size: 1000 entries
Default priority: 1
External cache groups:

Figure 8-68 Servlet Caching tab

5. Select the **Servlet Caching** tab and the **Enable Servlet Caching** box.

6. Click **OK** to close this window.
7. Click **Apply** to save the changes.
8. Repeat this procedure with each Application Server or ServerGroup that should use dynacache.
9. Restart the Application Servers or ServerGroups.

Define cachable servlets and JSPs

Here again, we have two possible ways to define the cachable servlets and JSPs: using a file, or when packaging the application.

Using a file for simple dynamic responses

The file is servletcache.xml.

1. If this file exists:

   $WAS_ROOT/properties/servletcache.xml

   WebSphere Application Server reads the definitions of the cachable servlets and JSPs from this file.
2. A sample file, servletcache.sample.xml, is provided and can be used as a template. To cache the showCfg servlet of the samples application for 100 seconds, it would look like this:

Example 8-3 Simple servletcache.xml

```
<?xml version="1.0" ?>
<!DOCTYPE servletCache SYSTEM "servletcache.dtd">
<servletCache>
  <servlet>
      <timeout seconds="100"/>
      <path uri="/webapp/examples/showCfg"/>
  </servlet>
</servletCache>
```

3. Edit the file and save it as servletcache.xml.

Using a file for dynamic query responses

You can also cache the responses of queries sent to a servlet (for example `WeatherDisplay.jsp?city=Miami`). Here you can specify which parameters should be used to generate the cache ID. With this configuration, servletcache.xml looks like this:

Example 8-4 Query servletcache.xml

```
<?xml version="1.0" ?>
<!DOCTYPE servletCache SYSTEM "servletcache.dtd">
<servletCache>
<servlet>
      <timeout seconds="100"/>
      <path uri="/webapp/examples/showCfg"/>
  </servlet>
  <servlet>
      <timeout seconds="60"/>
      <path uri="/WeatherProj/WeatherDisplay.jsp"/>
      <request>
          <attribute id="forecast" method="getCity" data_id="city"
required="true" />
      </request>
  </servlet>
</servletCache>
```

To enable programmatic invalidation of cache entries, we need to include the servlet that invalidates the entries but prevent it from being cached. The final servletcache.xml file is shown in Example 8-5.

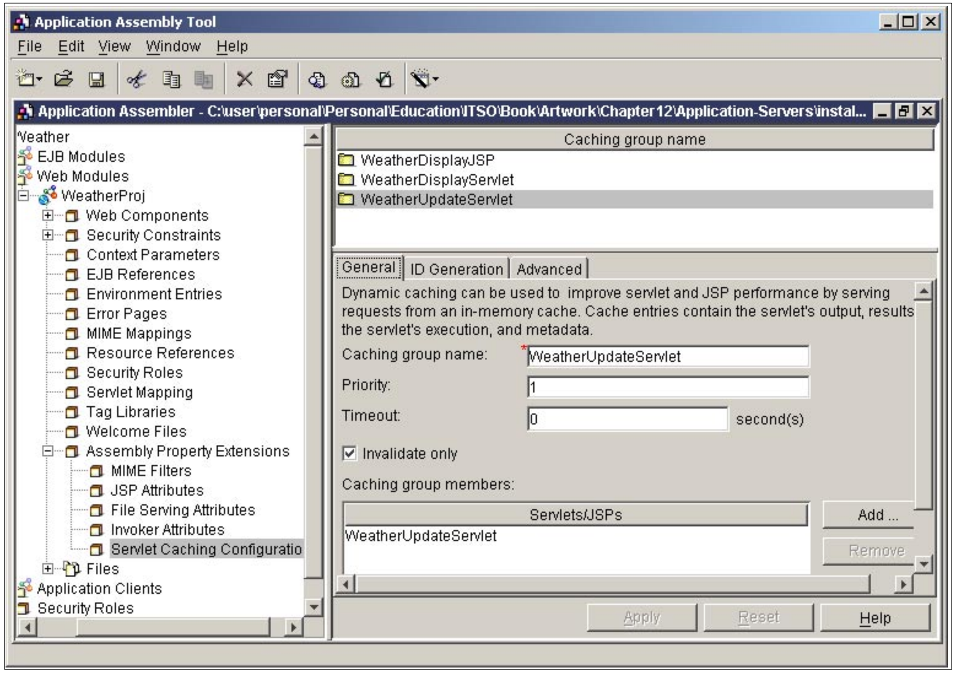
Example 8-5 Extended query servletcache.xml

```
<?xml version="1.0" ?>
<!DOCTYPE servletCache SYSTEM "servletcache.dtd">
<servletCache>
  <servlet>
      <timeout seconds="1800"/>
      <path uri="/webapp/examples/ping"/>
  </servlet>
  <servlet>
      <timeout seconds="100"/>
      <path uri="/webapp/examples/showCfg"/>
  </servlet>
  <servlet>
      <timeout seconds="60"/>
      <path uri="/WeatherProj/WeatherDisplay.jsp"/>
      <request>
          <attribute id="forecast" method="getCity" data_id="city"
required="true" />
      </request>
  </servlet>
  <servlet>
      <invalidateonly/>
      <path uri="/WeatherProj/weather"/>
      <request>
          <parameter id="update_city" invalidate="city" />
```

```
        </request>
    </servlet>
</servletCache>
```

Packaging the application

The other method to define cachable servlets and JSPs is to define the servlets as dynamically cachable when packaging the application. For example, the AAT (Application Assembly Tool) allows you to include the dynacache configuration for the servlets you chose to be cached. This configuration is written to the properties file ibm-web-ext.xmi. In Figure 8-69, you can see where this configuration is entered.



Caching group name
WeatherDisplayJSP
WeatherDisplayServlet
WeatherUpdateServlet

Dynamic caching can be used to improve servlet and JSP performance by serving requests from an in-memory cache. Cache entries contain the servlet's output, results the servlet's execution, and metadata.

Caching group name: WeatherUpdateServlet
Priority: 1
Timeout: 0 second(s)
[✓] Invalidate only
Caching group members:

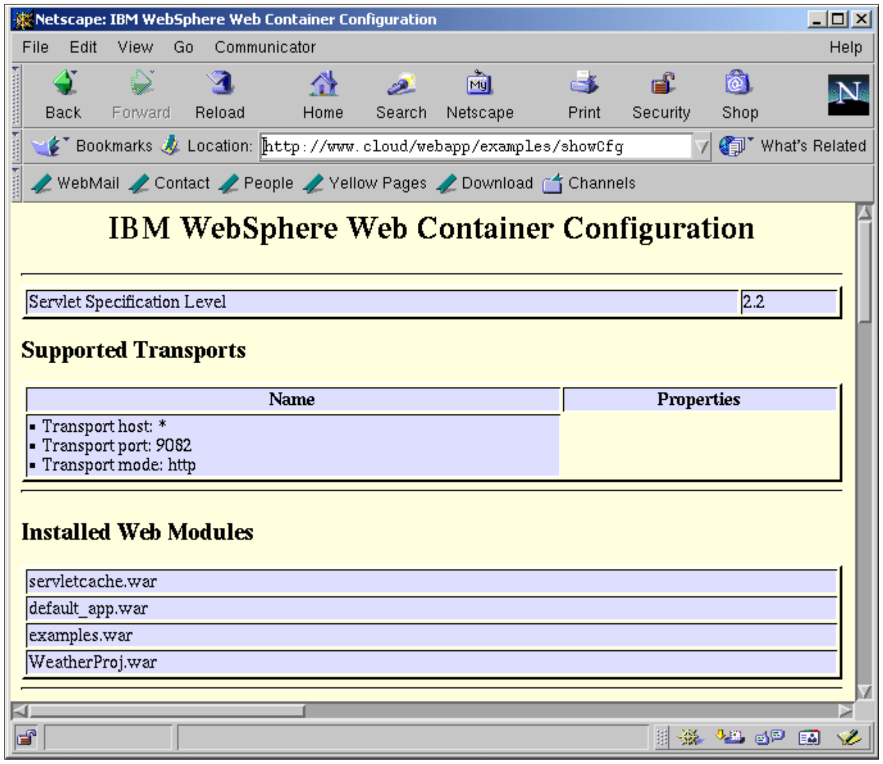
Servlets/JSPs
WeatherUpdateServlet

Figure 8-69 Application Assembly Tool

If you use this method, there will be no servletcache.xml file. This configuration file ibm-web-ext.xmi is contained within the application's *.ear file.

8.4 Testing local dynacache

1. Start the ServerGroup you want to test.
2. Start the samples application.
3. Start the ServletCacheMonitor application.


Netscape: IBM WebSphere Web Container Configuration
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop
Bookmarks Location: http://www.cloud/webapp/examples/showCfg What's Related
WebMail Contact People Yellow Pages Download Channels

IBM WebSphere Web Container Configuration

Servlet Specification Level	2.2

Supported Transports

Name	Properties
▪ Transport host: * ▪ Transport port: 9082 ▪ Transport mode: http	

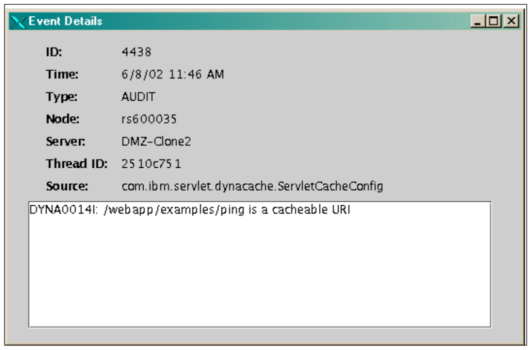
Installed Web Modules

servletcache.war
default_app.war
examples.war
WeatherProj.war

Figure 8-70 The showCfg servlet response

4. Open a browser and connect to the page as shown in Figure 8-70:

```
http://<www.cloud>/webapp/examples/showCfg
```



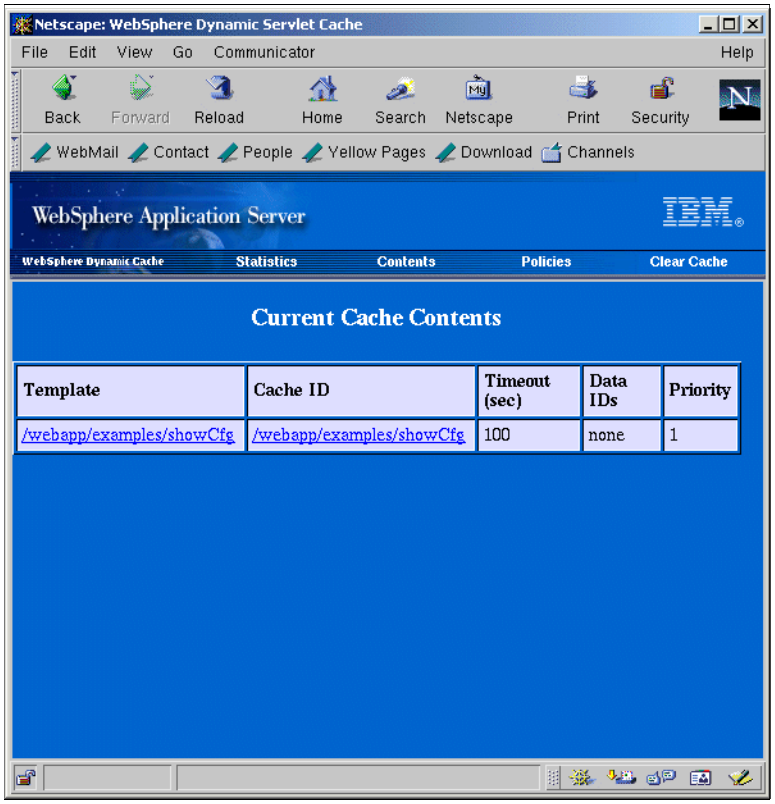
ID: 4438
Time: 6/8/02 11:46 AM
Type: AUDIT
Node: rs600035
Server: DMZ-Clone2
Thread ID: 2510c751
Source: com.ibm.servlet.dynacache.ServletCacheConfig

```
DYNA0014I: /webapp/examples/ping is a cacheable URI
```

Figure 8-71 Caching message

5. A message is issued when a cachable servlet is called the first time.
6. Load the page several times.
7. Open a second browser window and connect to the Cache Monitor:

   `http://<www.cloud>/servletcache/`
8. Click **Contents**.

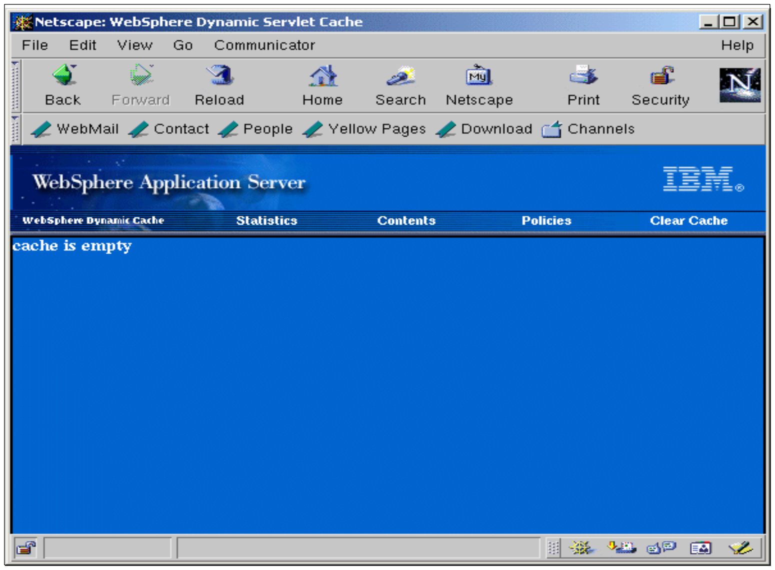

Netscape: WebSphere Dynamic Servlet Cache
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security
WebMail Contact People Yellow Pages Download Channels
WebSphere Application Server
IBM
WebSphere Dynamic Cache
Statistics
Contents
Policies
Clear Cache

Current Cache Contents

Template	Cache ID	Timeout (sec)	Data IDs	Priority
/webapp/examples/showCfg	/webapp/examples/showCfg	100	none	1

Figure 8-72 Servlet cache contents

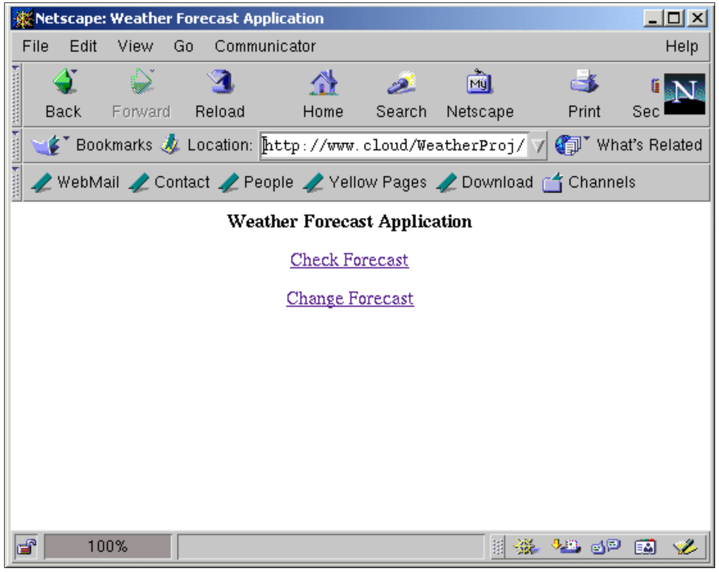
9. The content list shows the cached showCfg servlet. See Figure 8-72.
10. Click **Statistics** to see the number of cache hits.


Netscape: WebSphere Dynamic Servlet Cache
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security
WebMail Contact People Yellow Pages Download Channels
WebSphere Application Server
IBM
WebSphere Dynamic Cache
Statistics
Contents
Policies
Clear Cache

cache is empty

Figure 8-73 Cleared cache

11.After the timeout, the showCfg response is cleared from the cache.


Netscape: Weather Forecast Application
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Sec
Bookmarks Location: http://www.cloud/WeatherProj/ What's Related
WebMail Contact People Yellow Pages Download Channels
100%

Weather Forecast Application

Check Forecast

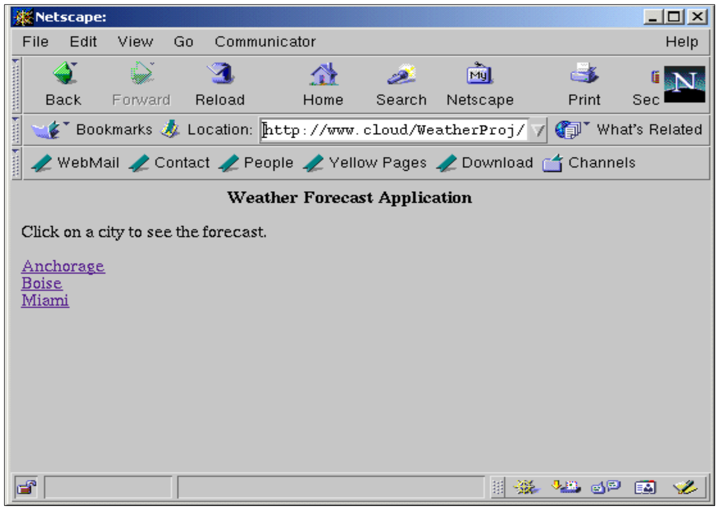
Change Forecast

Figure 8-74 The weather application

12. Now go to the weather application, as shown in Figure 8-74.

```
http://<www.cloud>/WeatherProj/index.html
```

13. Select **Check Forecast**.


Netscape:
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Sec
Bookmarks Location: http://www.cloud/WeatherProj/ What's Related
WebMail Contact People Yellow Pages Download Channels

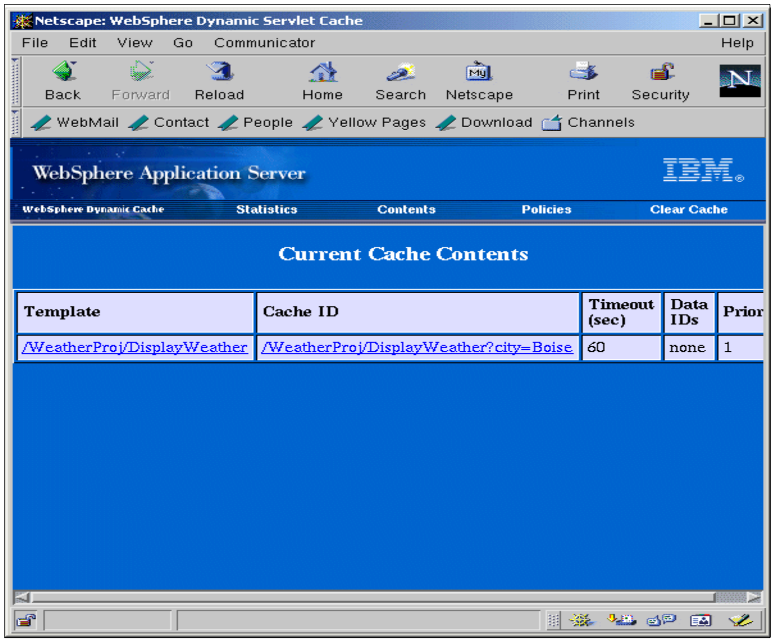
Weather Forecast Application

Click on a city to see the forecast.

Anchorage
Boise
Miami

Figure 8-75 Weather forecast application

14. Select a city for the forecast. We chose **Boise** (see Figure 8-75).

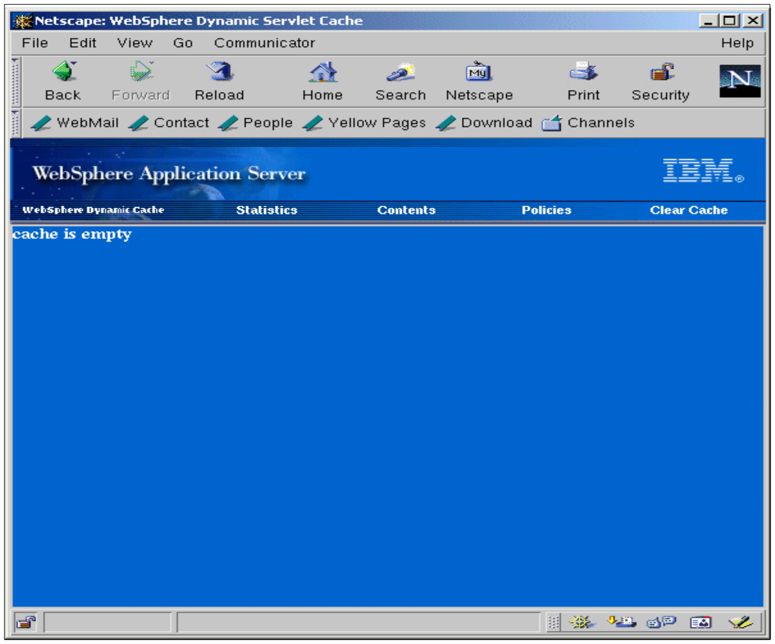

Netscape: WebSphere Dynamic Servlet Cache
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security
WebMail Contact People Yellow Pages Download Channels
WebSphere Application Server
IBM
WebSphere Dynamic Cache Statistics Contents Policies Clear Cache

Current Cache Contents

Template	Cache ID	Timeout (sec)	Data IDs	Prior
/WeatherProj/DisplayWeather	/WeatherProj/DisplayWeather?city=Boise	60	none	1

Figure 8-76 Cached query response

15. The servlet cache monitor shows the cached query response with the specified parameters.


Netscape: WebSphere Dynamic Servlet Cache
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security
WebMail Contact People Yellow Pages Download Channels
WebSphere Application Server
IBM
WebSphere Dynamic Cache
Statistics
Contents
Policies
Clear Cache

cache is empty

Figure 8-77 Servlet Cache Monitor

16. After the specified timeout, the entry is cleared from the cache.

8.5 Configure external dynacache

To enable the Caching Proxies to cache dynamic content, we need to install the dynacache subcomponent on the caching proxy machines. We also must configure the Caching Proxies and the Web Application Server nodes. We will add more information to the cachable servlet's definitions in the file servletcache.xml. Also, the local dynacache must be enabled on the Application Server nodes.

8.5.1 Caching proxy servers

On the caching proxy machines, we must install the dynacache subcomponent and configure some directives. These directives are configured by editing the ibmproxy.conf file.

1. Install dynamic caching support on the caching proxy servers. Dynamic caching support must be selected when the caching proxy is installed on the machine. Dynamic caching is a subcomponent of the Application Service at

the Edge component of WebSphere Edge Server V2. See Figure 6-8 on page 138 for details.

2. Edit the file ibmproxy.conf and change the proxy statements to proxyWAS statements.

   ```
   proxyWAS /* http://cloudhost:81/* www.cloud
   proxyWAS /* http://stormhost:81/* www.storm
   ```

3. Add the CacheQueries directives. Be careful to be specific enough to ensure that only the query results from servlets and JSPs you specified on the WebSphere Application Server node are cached!

   ```
   CacheQueries ALLWAYS http://cloudost:81/WeatherProj/*
   CacheQueries ALLWAYS http://stormost:81/WeatherProj/*
   ```

4. Add the ExternalCacheManager directive. Specify a name and global timeout for dynamic cache entries.

   ```
   ExternalCacheManager WAS-DynaManager 10 minutes
   ```

5. Find the line:

   ```
   # ===== JSP Plug-in =====
   ```

   and uncomment the line following it. It is the service directive:

   ```
   Service /WES_External_Adapter C:\Progra~1\IBM\edge\cp\lib\plugins\
   dynacache\dyna_plugin.dll:exec_dynacmd
   ```

   Note: The line above should read as one single line.

6. Restart the caching proxy.

8.5.2 Web Application Servers

Copy dynacache archives

Make the dynacache archives supplied with the caching proxy available to WebSphere Application Server.

Copy the wses_*.jar files from a Caching Proxy node to the $WAS_ROOT/lib directories on each Web Application Server node. The jar files are found in the following directory:

Unix: /opt/ibm/edge/asate/lib

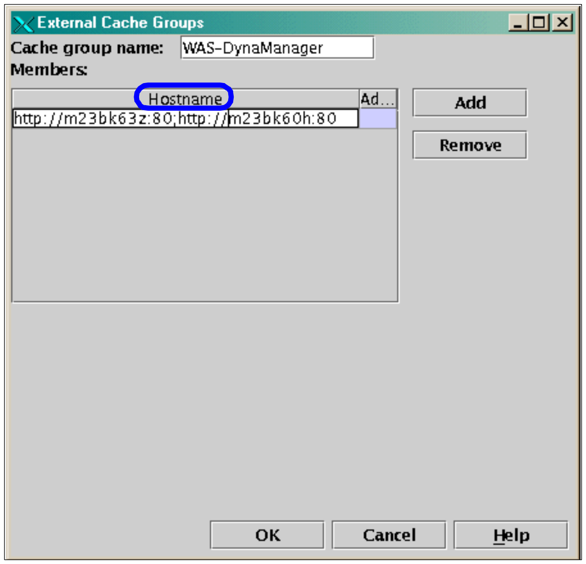
Windows: C:\Progra~1\IBM\edge\asate\lib

Configure WebSphere for an External Cache Group

Once again, there are two possible ways to complete this task. You may use the GUI (see below), or you may edit an *.xml file (see “Using the XML config file” on page 291).

Using the GUI

1. Open the WebSphere Administrative Console.
2. Select an Application Server or a ServerGroup in the left-hand pane.
3. Select the **Web Container Service** in the right hand pane.
4. Click **Edit**.
5. Select the **Servlet Caching** tab.
6. Click **Add an External Cache Group**.



Cache group name: WAS-DynaManager

Members:

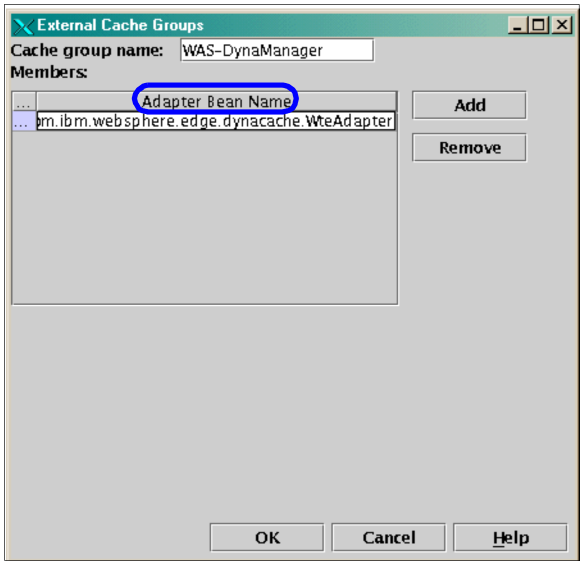
Hostname	Ad...
http://m23bk63z:80;http://m23bk60h:80	

Figure 8-78 External Cache Group properties

7. In the Hostname field, enter caching proxy nodes in the format:

   ```
   <protocol>://<hostname>:<port>
   ```

   Separate multiple proxy servers with a semicolon. Refer to Figure 8-78.


External Cache Groups

Cache group name: WAS-DynaManager

Members:

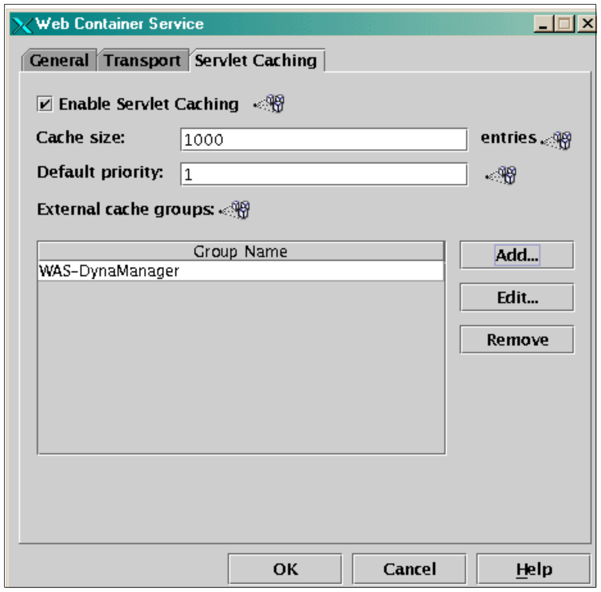
...	Adapter Bean Name
...	om.ibm.websphere.edge.dynacache.WteAdapter

Figure 8-79 Dynacache adapter

8. In the Adapter Bean Name field enter:

   `com.ibm.websphere.edge.dynacache.WteAdapter`

9. Click **OK** to save the configuration.


Web Container Service

☑ Enable Servlet Caching

Cache size: 1000 entries

Default priority: 1

External cache groups:

Group Name
WAS-DynaManager

Figure 8-80 The new External Cache Group

10. Click **OK** and make sure to click **Apply** to save the changes.

Using the XML config file

1. On the Web Application Servers, create the following directory path:

   Unix: /opt/ibm/edge/asate/conf

   Windows: C:\Program Files\IBM\edge\asate\conf

2. Copy the file dynaedge-cfg.xml from the same directory on the caching proxy nodes to the new directories on the Web Application Server nodes.
3. Specify the External Cache Group and add the Proxy Server nodes as members to it. You can see our two Caching Proxies in Example 8-6.

Example 8-6 dynaedge-cfg.xml

```
<?xml version="1.0"?>
<!DOCTYPE EdgeServerCfg [

   <!ELEMENT EdgeServerCfg (EdgeServer* | MasterCDS* | ConnectionProperties?)>
   <!ELEMENT EdgeServer EMPTY>
   <!ATTLIST EdgeServer
      endpointCDATA #REQUIRED
      user      CDATA #IMPLIED
      userPasswdCDATA #IMPLIED
      invalidation-urlCDATA #IMPLIED
      URI-type(relative | absolute) "relative"
   >

   <!ELEMENT MasterCDS EMPTY>
   <!ATTLIST MasterCDS
      endpoint  CDATA #REQUIRED
      domain-nameCDATA #IMPLIED
   >

   <!ELEMENT ConnectionProperties EMPTY>
   <!ATTLIST ConnectionProperties
      timeOut   CDATA #REQUIRED
      certFileCDATA #IMPLIED
      certPasswdCDATA #IMPLIED
      certFileTypeCDATA #IMPLIED
   >
]>

<EdgeServerCfg>
    <EdgeServer
   endpoint = "http://m23bk63z:8080"
        user = "wsesadmin"
   userPasswd = "wsesadmin"
   invalidation-url = "/WES_External_Adapter"
   URI-type= "absolute"
    />
    <EdgeServer
   endpoint = "http://m23bk60h:8080"
        user = "wsesadmin"
   userPasswd = "wsesadmin"
   invalidation-url = "/WES_External_Adapter"
   URI-type= "absolute"
    />
    <ConnectionProperties timeout = "60" />
</EdgeServerCfg>
```

Modify the definitions for the cachable servlets and JSPs

When an External Cache Group comes into play, the mechanism to invalidate the cached servlets and JSPs again must be involved. The Java archives copied from the caching proxy node provide the classes to do this job. We only have to define the classes in the servletcache.xml file. In each definition of a cachable servlet or JSP, we add two objects. One object specifies the class of the Metadatagenerator. The other object specifies the name of the External Cache Group.

Simple servlet definition

Examine the changed servletcache.xml. Compare Example 8-7 with Example 8-3 on page 278.

Example 8-7 Simple servlet definition

```
<?xml version="1.0" ?>
<!DOCTYPE servletCache SYSTEM "servletcache.dtd">
<servletCache>
  <servlet>
      <timeout seconds="1800"/>
      <metadatagenerator
class="com.ibm.websphere.edge.dynacache.WteMetaDataGeneratorImpl"/>
      <externalcache id="WAS-DynaManager"/>
      <path uri="/webapp/examples/ping"/>
  </servlet>
</servletCache>
```

Query results definition

The cache ID of query results is always generated using all available parameters. Therefore, you cannot specify them in the servlet definitions. The only exemptions are for `invalidateonly` statements. The servletcache.xml for external caching of the WeatherDisplay.jsp would have entries as shown in Example 8-8.

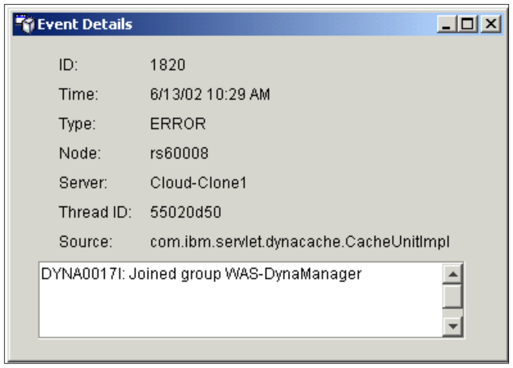
Example 8-8 Query Response definition

```
<?xml version="1.0" ?>
<!DOCTYPE servletCache SYSTEM "servletcache.dtd">
<servletCache>
  <servlet>
      <timeout seconds="100"/>
      <path uri="/webapp/examples/showCfg"/>
      <metadatagenerator
class="com.ibm.websphere.edge.dynacache.WteMetaDataGeneratorImpl"/>
      <externalcache id="WAS-DynaManager"/>
  </servlet>
  <servlet>
      <timeout seconds="60"/>
      <path uri="/WeatherProj/WeatherDisplay.jsp"/>
      <metadatagenerator
class="com.ibm.websphere.edge.dynacache.WteMetaDataGeneratorImpl"/>
      <externalcache id="WAS-DynaManager"/>
  </servlet>
</servletCache>
```

As you can see, there is no difference in the definition of simple or query results servlets and JSPs.

8.6 Testing external dynacache

1. Start both ServerGroups through the Admin Console by either selecting a ServerGroup and clicking the **start** icon, or right-clicking the ServerGroup and selecting **Start**. There will be an error message in the messages pane which is not really an error. Click **Details** to see the message. It is shown in Figure 8-81.


Event Details

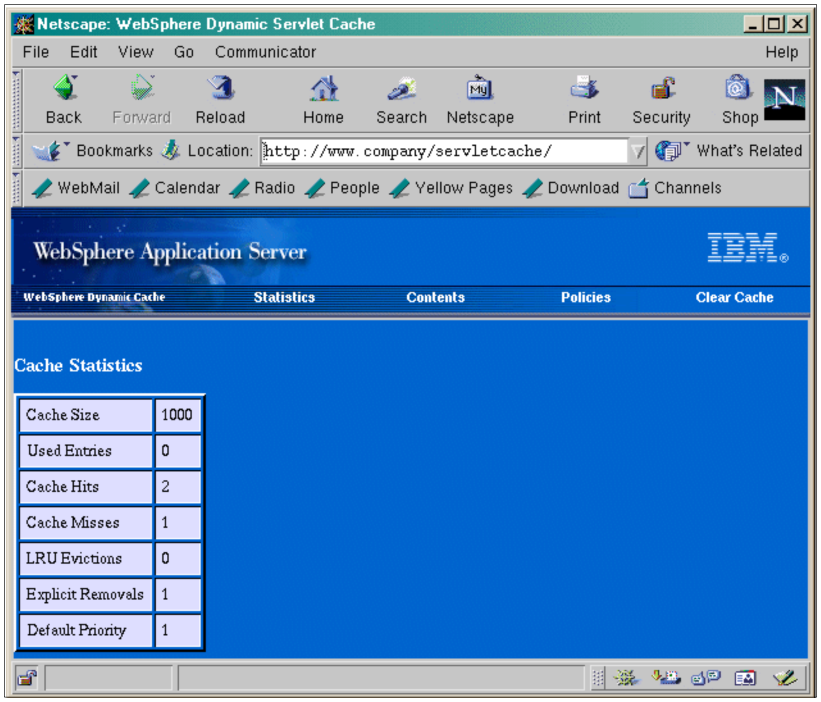
ID:	1820
Time:	6/13/02 10:29 AM
Type:	ERROR
Node:	rs60008
Server:	Cloud-Clone1
Thread ID:	55020d50
Source:	com.ibm.servlet.dynacache.CacheUnitImpl

```
DYNA0017I: Joined group WAS-DynaManager
```

Figure 8-81 Wrong error message

2. Open a browser and go to the /webapp/examples/showCfg servlet.
3. Open a new Browser window and connect to the servlet cache monitor:

   `http://www.cloud/servletcache/`


Netscape: WebSphere Dynamic Servlet Cache
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop
Bookmarks Location: http://www.company/servletcache/ What's Related
WebMail Calendar Radio People Yellow Pages Download Channels
WebSphere Application Server
IBM
WebSphere Dynamic Cache
Statistics
Contents
Policies
Clear Cache

Cache Statistics

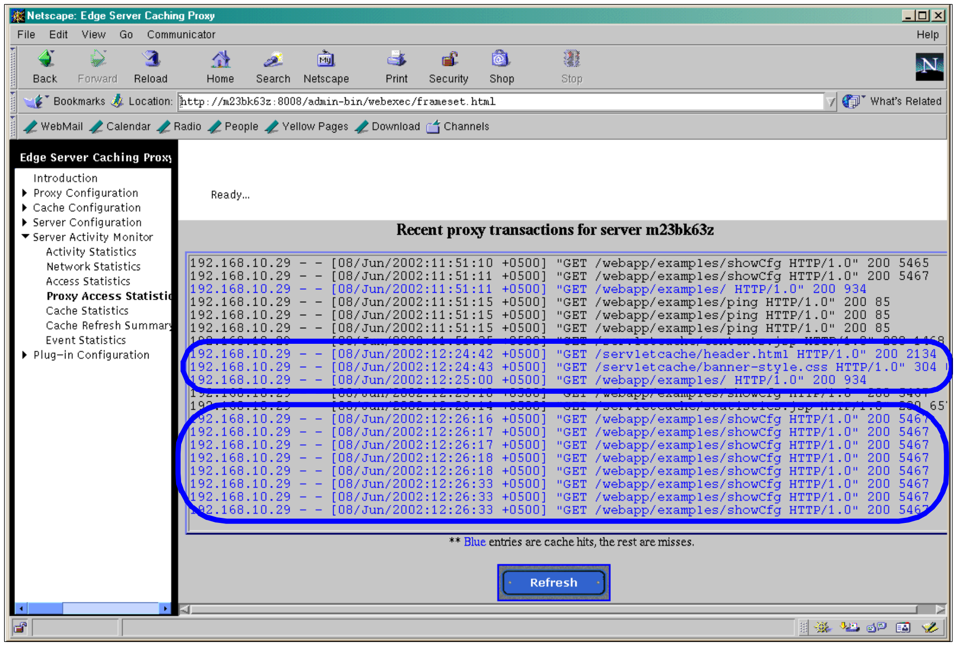
Cache Size	1000
Used Entries	0
Cache Hits	2
Cache Misses	1
LRU Evictions	0
Explicit Removals	1
Default Priority	1

Figure 8-82 Cache monitor shows one cache miss and one entry

4. Load the showCfg servlet several times.
5. Open one more browser window and connect to the caching proxy:

   `http://www.cloud/wses/Frntpage.html`

6. Log on to the Administration Forms.
7. Select **Server Activity Monitor**.
8. Select **Proxy Access Statistics**.


Netscape: Edge Server Caching Proxy
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: http://m23bk63z:8008/admin-bin/webexec/frameset.html
What's Related
WebMail Calendar Radio People Yellow Pages Download Channels
Edge Server Caching Proxy
Introduction
Proxy Configuration
Cache Configuration
Server Configuration
Server Activity Monitor
Activity Statistics
Network Statistics
Access Statistics
Proxy Access Statistic
Cache Statistics
Cache Refresh Summary
Event Statistics
Plug-in Configuration
Ready...
Refresh

Recent proxy transactions for server m23bk63z

```
192.168.10.29 - - [08/Jun/2002:11:51:10 +0500] "GET /webapp/examples/showCfg HTTP/1.0" 200 5465
192.168.10.29 - - [08/Jun/2002:11:51:11 +0500] "GET /webapp/examples/showCfg HTTP/1.0" 200 5467
192.168.10.29 - - [08/Jun/2002:11:51:11 +0500] "GET /webapp/examples/ HTTP/1.0" 200 934
192.168.10.29 - - [08/Jun/2002:11:51:15 +0500] "GET /webapp/examples/ping HTTP/1.0" 200 85
192.168.10.29 - - [08/Jun/2002:11:51:15 +0500] "GET /webapp/examples/ping HTTP/1.0" 200 85
192.168.10.29 - - [08/Jun/2002:11:51:15 +0500] "GET /webapp/examples/ping HTTP/1.0" 200 85
192.168.10.29 - - [08/Jun/2002:11:51:35 +0500] "GET /servletcache/contents.jsp HTTP/1.0" 200 1468
192.168.10.29 - - [08/Jun/2002:12:24:42 +0500] "GET /servletcache/header.html HTTP/1.0" 200 2134
192.168.10.29 - - [08/Jun/2002:12:24:43 +0500] "GET /servletcache/banner-style.css HTTP/1.0" 304
192.168.10.29 - - [08/Jun/2002:12:25:00 +0500] "GET /webapp/examples/ HTTP/1.0" 200 934
192.168.10.29 - - [08/Jun/2002:12:25:18 +0500] "GET /webapp/examples/showCfg HTTP/1.0" 200 5467
192.168.10.29 - - [08/Jun/2002:12:26:14 +0500] "GET /servletcache/statistics.jsp HTTP/1.0" 200 65
192.168.10.29 - - [08/Jun/2002:12:26:16 +0500] "GET /webapp/examples/showCfg HTTP/1.0" 200 5467
192.168.10.29 - - [08/Jun/2002:12:26:17 +0500] "GET /webapp/examples/showCfg HTTP/1.0" 200 5467
192.168.10.29 - - [08/Jun/2002:12:26:17 +0500] "GET /webapp/examples/showCfg HTTP/1.0" 200 5467
192.168.10.29 - - [08/Jun/2002:12:26:18 +0500] "GET /webapp/examples/showCfg HTTP/1.0" 200 5467
192.168.10.29 - - [08/Jun/2002:12:26:18 +0500] "GET /webapp/examples/showCfg HTTP/1.0" 200 5467
192.168.10.29 - - [08/Jun/2002:12:26:33 +0500] "GET /webapp/examples/showCfg HTTP/1.0" 200 5467
192.168.10.29 - - [08/Jun/2002:12:26:33 +0500] "GET /webapp/examples/showCfg HTTP/1.0" 200 5467
192.168.10.29 - - [08/Jun/2002:12:26:33 +0500] "GET /webapp/examples/showCfg HTTP/1.0" 200 546
```

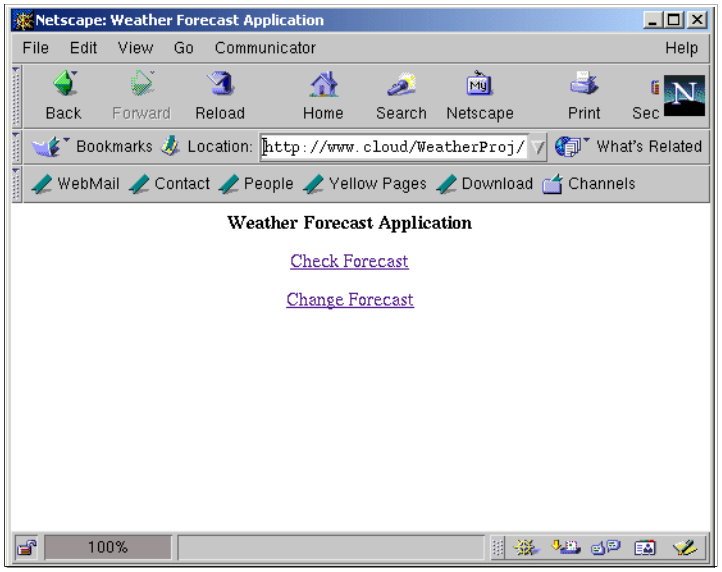
** Blue entries are cache hits, the rest are misses.

Figure 8-83 Proxy Access Log

9. The Proxy Access Statistics show that the showCfg servlet was called once only and then served from cache (blue entries).

10. To test the weather application, connect to the following URL.

    `http://www.cloud/WeatherProj/index.html`


Netscape: Weather Forecast Application
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Sec
Bookmarks Location: http://www.cloud/WeatherProj/ What's Related
WebMail Contact People Yellow Pages Download Channels
100%

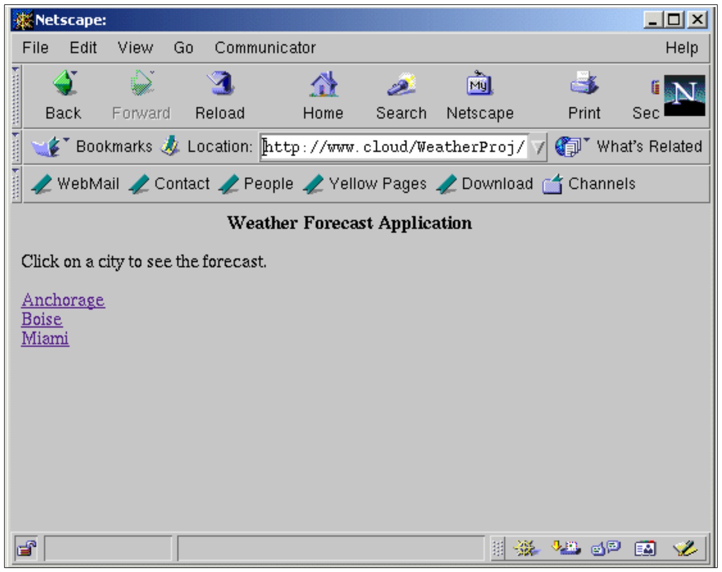
Weather Forecast Application

Check Forecast

Change Forecast

Figure 8-84 The weather application

11.Select **Check Forecast**.


Netscape:
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Sec
Bookmarks Location: http://www.cloud/WeatherProj/ What's Related
WebMail Contact People Yellow Pages Download Channels

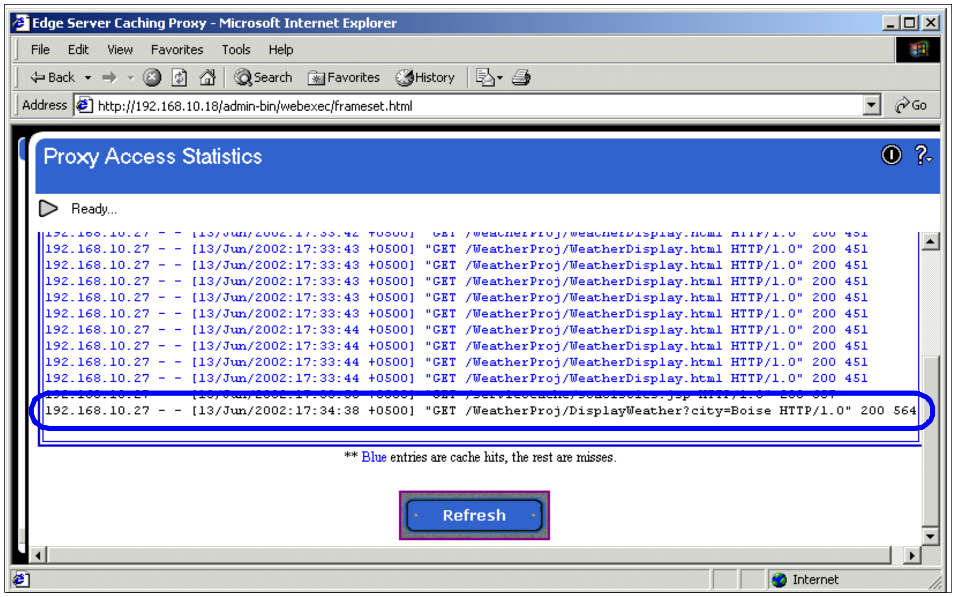
Weather Forecast Application

Click on a city to see the forecast.

Anchorage
Boise
Miami

Figure 8-85 City selection

12.Select a city for which to get a forecast.

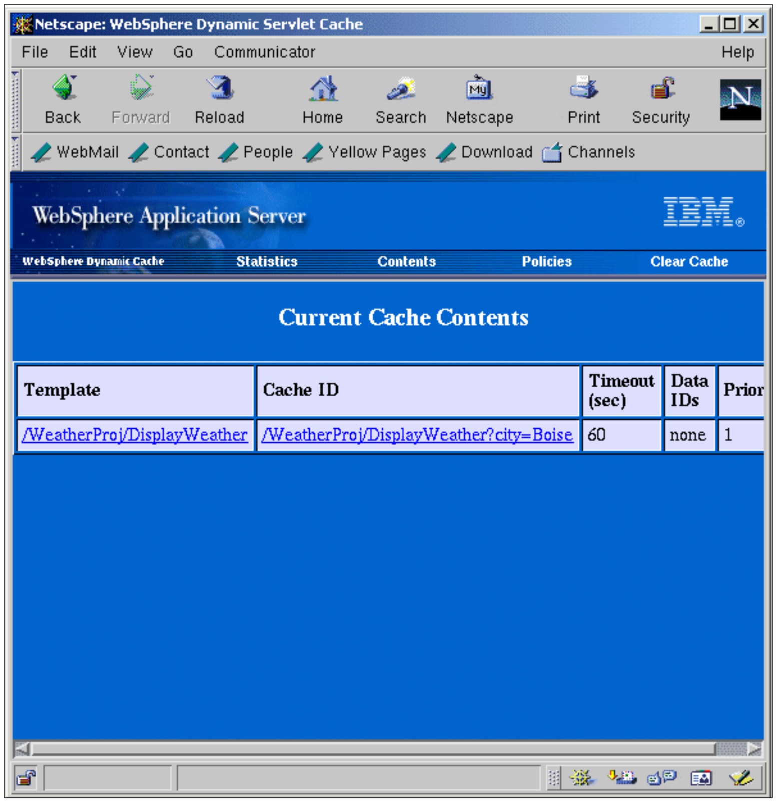

Edge Server Caching Proxy - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://192.168.10.18/admin-bin/webexec/frameset.html
Go
Proxy Access Statistics
Ready...
Refresh
Internet

```
192.168.10.27 - - [13/Jun/2002:17:33:42 +0500] "GET /WeatherProj/WeatherDisplay.html HTTP/1.0" 200 451
192.168.10.27 - - [13/Jun/2002:17:33:43 +0500] "GET /WeatherProj/WeatherDisplay.html HTTP/1.0" 200 451
192.168.10.27 - - [13/Jun/2002:17:33:43 +0500] "GET /WeatherProj/WeatherDisplay.html HTTP/1.0" 200 451
192.168.10.27 - - [13/Jun/2002:17:33:43 +0500] "GET /WeatherProj/WeatherDisplay.html HTTP/1.0" 200 451
192.168.10.27 - - [13/Jun/2002:17:33:43 +0500] "GET /WeatherProj/WeatherDisplay.html HTTP/1.0" 200 451
192.168.10.27 - - [13/Jun/2002:17:33:43 +0500] "GET /WeatherProj/WeatherDisplay.html HTTP/1.0" 200 451
192.168.10.27 - - [13/Jun/2002:17:33:44 +0500] "GET /WeatherProj/WeatherDisplay.html HTTP/1.0" 200 451
192.168.10.27 - - [13/Jun/2002:17:33:44 +0500] "GET /WeatherProj/WeatherDisplay.html HTTP/1.0" 200 451
192.168.10.27 - - [13/Jun/2002:17:33:44 +0500] "GET /WeatherProj/WeatherDisplay.html HTTP/1.0" 200 451
192.168.10.27 - - [13/Jun/2002:17:33:44 +0500] "GET /WeatherProj/WeatherDisplay.html HTTP/1.0" 200 451
[illegible]
192.168.10.27 - - [13/Jun/2002:17:34:38 +0500] "GET /WeatherProj/DisplayWeather?city=Boise HTTP/1.0" 200 564
```

** Blue entries are cache hits, the rest are misses.

Figure 8-86 Proxy Access Log after forecast

13. On the first request, the proxy server gets the response from the application server.


Netscape: WebSphere Dynamic Servlet Cache
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security
WebMail Contact People Yellow Pages Download Channels
WebSphere Application Server
IBM
WebSphere Dynamic Cache Statistics Contents Policies Clear Cache

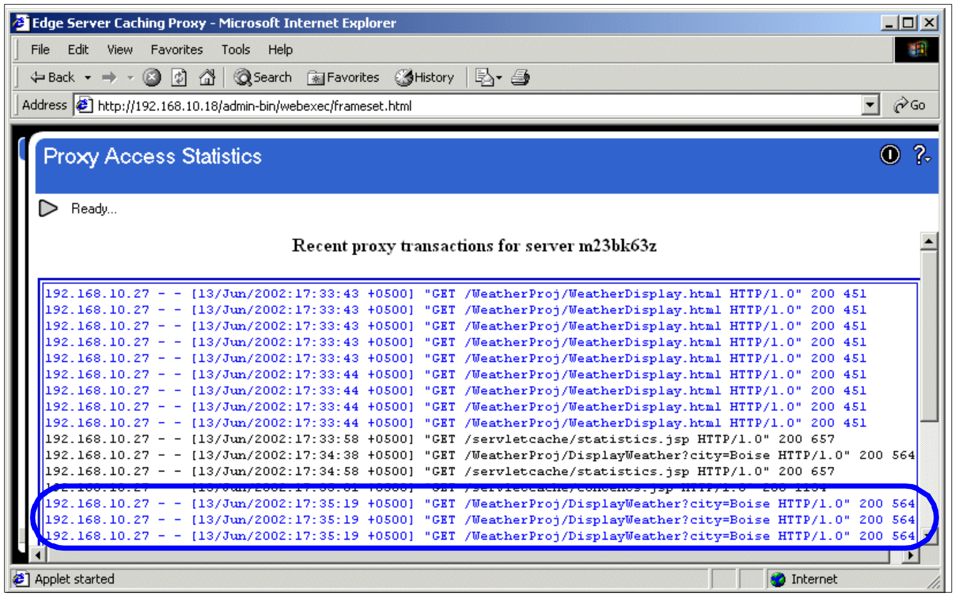
Current Cache Contents

Template	Cache ID	Timeout (sec)	Data IDs	Prior
/WeatherProj/DisplayWeather	/WeatherProj/DisplayWeather?city=Boise	60	none	1

Figure 8-87 Cache monitor

14.On the application server, the response has been cached.

15.Request the forecast for the same city multiple times.

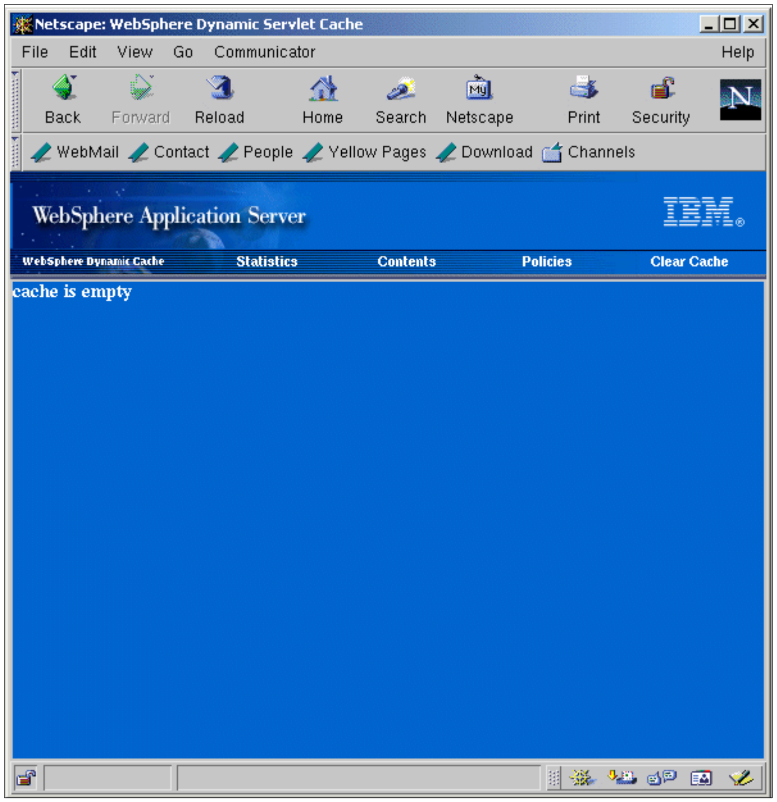

Edge Server Caching Proxy - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://192.168.10.18/admin-bin/webexec/frameset.html
Go
Proxy Access Statistics
Ready...
Applet started
Internet

Recent proxy transactions for server m23bk63z

```
192.168.10.27 - - [13/Jun/2002:17:33:43 +0500] "GET /WeatherProj/WeatherDisplay.html HTTP/1.0" 200 451
192.168.10.27 - - [13/Jun/2002:17:33:43 +0500] "GET /WeatherProj/WeatherDisplay.html HTTP/1.0" 200 451
192.168.10.27 - - [13/Jun/2002:17:33:43 +0500] "GET /WeatherProj/WeatherDisplay.html HTTP/1.0" 200 451
192.168.10.27 - - [13/Jun/2002:17:33:43 +0500] "GET /WeatherProj/WeatherDisplay.html HTTP/1.0" 200 451
192.168.10.27 - - [13/Jun/2002:17:33:43 +0500] "GET /WeatherProj/WeatherDisplay.html HTTP/1.0" 200 451
192.168.10.27 - - [13/Jun/2002:17:33:44 +0500] "GET /WeatherProj/WeatherDisplay.html HTTP/1.0" 200 451
192.168.10.27 - - [13/Jun/2002:17:33:44 +0500] "GET /WeatherProj/WeatherDisplay.html HTTP/1.0" 200 451
192.168.10.27 - - [13/Jun/2002:17:33:44 +0500] "GET /WeatherProj/WeatherDisplay.html HTTP/1.0" 200 451
192.168.10.27 - - [13/Jun/2002:17:33:44 +0500] "GET /WeatherProj/WeatherDisplay.html HTTP/1.0" 200 451
192.168.10.27 - - [13/Jun/2002:17:33:58 +0500] "GET /servletcache/statistics.jsp HTTP/1.0" 200 657
192.168.10.27 - - [13/Jun/2002:17:34:38 +0500] "GET /WeatherProj/DisplayWeather?city=Boise HTTP/1.0" 200 564
192.168.10.27 - - [13/Jun/2002:17:34:58 +0500] "GET /servletcache/statistics.jsp HTTP/1.0" 200 657
192.168.10.27 - - [13/Jun/2002:17:35:01 +0500] "GET /servletcache/contents.jsp HTTP/1.0" 200 1134
192.168.10.27 - - [13/Jun/2002:17:35:19 +0500] "GET /WeatherProj/DisplayWeather?city=Boise HTTP/1.0" 200 564
192.168.10.27 - - [13/Jun/2002:17:35:19 +0500] "GET /WeatherProj/DisplayWeather?city=Boise HTTP/1.0" 200 564
192.168.10.27 - - [13/Jun/2002:17:35:19 +0500] "GET /WeatherProj/DisplayWeather?city=Boise HTTP/1.0" 200 564
```

Figure 8-88 Proxy Access Log

16.The Proxy Access Log shows the requests being served from cache.


Netscape: WebSphere Dynamic Servlet Cache
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security
WebMail Contact People Yellow Pages Download Channels
WebSphere Application Server
IBM
WebSphere Dynamic Cache
Statistics
Contents
Policies
Clear Cache

cache is empty

Figure 8-89 Local dynacache monitor

17. After the timeout, the cache entry is removed from cache.

8.6.1 Dynamic query responses issue

At the time that this book was being written, the caching proxy dynacache feature worked as expected (with invalidation messages coming from WebSphere Application Server) when simple URLs with JSPs and dynamic pages were used.

However, if the cached dynamic page includes query results (for example, `cache id = /WeatherProj/WeatherDisplay.jsp?city=Boise`), it appears that the external cached adapter within WebSphere Application Server does not send invalidation messages when the timeout value is reached. However, the query responses are removed correctly by the caching proxy when the global timeout for an External Cache Group expires.

In addition, if the servlet has been cached without a query once (without the `?city=.....`) then everything works as expected.

9 Company environment

In this sample environment, we consider the need for better protection for certain kinds of data. Although security implementation is not a subject of this book, this runtime topology provides the architecture to enforce it. It is based on 3.4.2, “Basic Runtime pattern variation 6: separation” on page 50.

In this environment, the application server functionalities are divided into separate presentation and application nodes. The presentation nodes still reside in the DMZ, while the application or business logic is behind the domain firewall in the secure network. This way, a user can be challenged for authentication before a request is passed on into the secure part of your company’s network. We do implement a proxy server layer in the DMZ. It is reasonable to cache content such as home page, pictures and other static content. But you have to be careful about what is being cached because access to the proxy servers is not secure. In this situation, where you are the owner of both the caching proxy servers and the content, you are able to control what is to be cached and what is not.

Another possibility is to use an Authenticating Proxy node. It intercepts all incoming requests for defined resources and enforces authentication. In such an environment, all content servers can be placed in the secure network. Access to the proxy servers’ caches is also secured in this setup. The WebSphere Edge Server Tivoli Access Manager plug-in and LDAP plug-in both provide the product to implement such a solution on different levels. The Tivoli Access Manager plug-in is discussed in 5.7, “Authenticating proxy” on page 126.

9.1 Scenario description

In this scenario, we use an online trading application to demonstrate mutual high availability load balancers and the benefits of dividing the presentation and business logic into separate nodes.

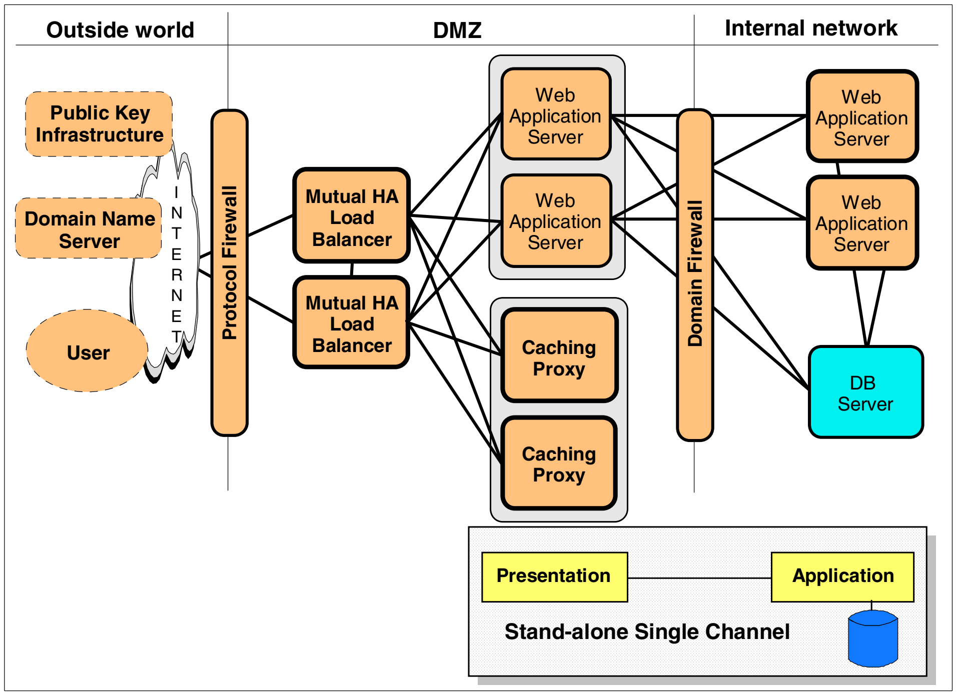


Figure 9-1 Sample company environment

In this configuration, both mutual high availability load balancer nodes are responsible for load balancing requests across servers in their respective clusters. In addition, each load balancer backs up the other load balancer by taking over cluster responsibility in case of a failure. Refer to 4.2.6, “Load Balancer Mutual High Availability feature” on page 73 for a further discussion of mutual high availability.

9.1.1 Traffic flow

Refer to Figure 9-1 when studying the following request/response traffic flow.

Request

1. The virtual www addresses are all mapped to the same IP cluster address and arrive at the mutual high availability load balancer node configured to

support the virtual www addresses. The load balancer, using the MAC forwarding method, does not examine the contents of the request header. It distributes requests between the caching proxy nodes without changing the request.

2. The caching proxies in turn forward the requests that cannot be served from cache. Depending on the hostname contained in the request header, they forward it to the virtual host address that matches the mapping rule (the caching proxy machines change the hostname in the request header and the IP address of the packet). Because these virtual host names are again mapped to an IP cluster address on one of the mutual high availability load balancer nodes, they are sent there, but use a different cluster address.
3. The packet is processed through the load balancer node. The load balancer sends the request to the WebSphere Application Server nodes.
4. The request header now contains the hostname of the caching proxy (inserted by the caching proxy). The WebSphere plug-in on the WebSphere Application Server decides, according to a match of this hostname and the requested URI, to which ServerGroup the request belongs and distributes the requests between the clones of this ServerGroup. It changes the IP address of the packet to the IP address of the transport host for this clone. The hostname in the header remains unchanged. Because each clone on a given host has a unique port number, it will connect to the right one and send the packet.
5. Each clone in turn has its own HTTP daemon, which receives the packet. The Web Container that owns the application context root that is found in the URI will receive the request.
6. The Web Container determines the right servlet instance to process the request and forwards it for processing.
7. If the servlet has to call an EJB at the back-end Application Server, it queries the naming service for its JNDI name and sends a request to the EJB.

Response

8. The servlet processes the request and incorporates response data from EJB calls into the response (probably into a JSP).
9. The servlet response is given back to the HTTP daemon of the clone.
10. The clone’s HTTP daemon swaps the source and the destination IP addresses of the packet and sends it back on its way. This is how any server responds to a client on IP-based traffic.
11. The caching proxy receives the packet and remembers the original client IP address. It puts this IP address as the destination and itself as the source IP address of the packet, then sends it back to the client.

9.1.2 Scalability

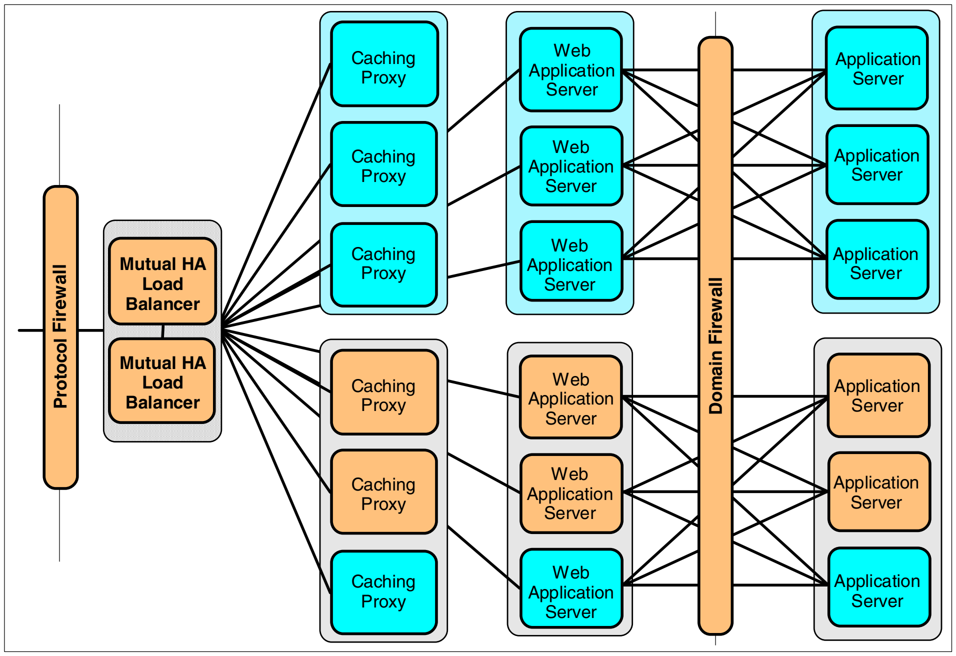


Figure 9-2 Scalability

Full scalability is provided at any level. Servers can be added to any cluster and even whole clusters could be added to extend the configuration. Also, each application can be scaled to its needs by adding more clones to its ServerGroup or even creating an additional ServerGroup for it.

9.1.3 Naming

Note that in this scenario, the virtual hosts again are name-based virtual hosts. They are all based on a single IP address. This makes it very easy to extend. There is no need for additional an IP address when a new hostname is defined. And even more important, the load balancer configuration never needs to be changed. Simply add a new definition to DNS or the hosts files.

Table 9-1 Naming in the example company environment

Item	name
Internet virtual Host	www.company
IP Address	192.168.10.80

Item	name
Back-end virtual Host	companyhost:82
IP Address	192.168.10.90
Trade Database name	tradedb
Trade Data Source name	trade_DS

9.1.4 Node functionalities

Table 9-2 Machines used in the company environment scenario

Host Name	IP Address	OS and Software	Service
23VNX85	192.168.10.14	Windows 2000 WebSphere Edge Server V2.0	Mutual high availability Load Balancer
23VNX72	192.168.10.15	Windows 2000 WebSphere Edge Server V2.0	Mutual high availability Load Balancer
M23BK63Z	192.168.10.18	WebSphere Edge Server V2.0	Caching Proxy
M23BK60H	192.168.10.4	WebSphere Edge Server V2.0	Caching Proxy
RS60008	192.168.10.27	AIX 4.3.3 IBM WebSphere Application Server V4.0.3 IBM HTTP Server V 1.3.19	Presentation Web Application Server
RS600035	192.168.10.29	AIX 4.3.3 IBM WebSphere Application Server V4.0.3 IBM HTTP Server V 1.3.19	Presentation Web Application Server
M23VNX64	9.24.105.112	Windows 2000 IBM WebSphere Application Server V4.0.3	Application Web Application Server

Host Name	IP Address	OS and Software	Service
M23X2900	9.24.105.121	Windows 2000 IBM WebSphere Application Server V4.0.3	Application Web Application Server
M23VNX64	9.24.105.112	Windows 2000 IBM DB2 V7.2	Database Server

The whole scenario can be realized with nine nodes: two load balancer nodes, two caching proxy nodes, four WebSphere Application Server and one database server node. The databases could probably be created on an existing database server.

9.2 Configuration description

9.2.1 Name resolution

In our sample trading environment, we configured the following mappings.

Example 9-1 Sample hosts file

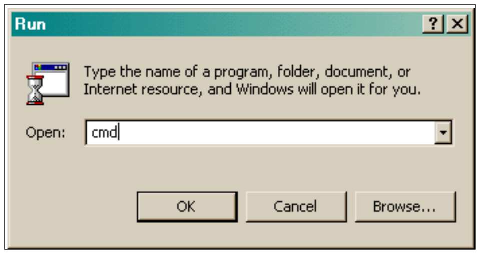
```
# Sample hosts file for Company-Environment SG24-6822-00
#============================================================================

127.0.0.1       localhost

# Server Host Names
#============================================================================

192.168.10.14   m23vnx85        # Mutual high availability Load Balancer
192.168.10.15   m23vnx72        # Mutual high availability Load Balancer
192.168.10.18   m23bk63z        # Caching Proxy  1
192.168.10.4    m23bk60h        # Caching Proxy  2
192.168.10.27   rs60008         # Content Server 1 (WAS)
192.168.10.29   rs600035        # Content Server 2 (WAS)
9.24.105.112    m23vnx64        # Content Server 3 (WAS)
9.24.105.121    m23x2900        # Content Server 4 (WAS)
9.24.105.112    m23vnx64        # Backend Database Server

# virtual Host Names
#============================================================================

192.168.10.80   www.company     # Virtual www address for the Company
192.168.10.90   companyhost     # Virtual address for the Content Servers
```

9.2.2 Database server

With the trade application comes a script which creates the database and the tables within it. Our database server is a Windows machine, so we will use the tradeDB.bat file. Copy it to a directory on your database server.

Create the database

1. Log on as the instance owner, db2admin in our case.



Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: cmd

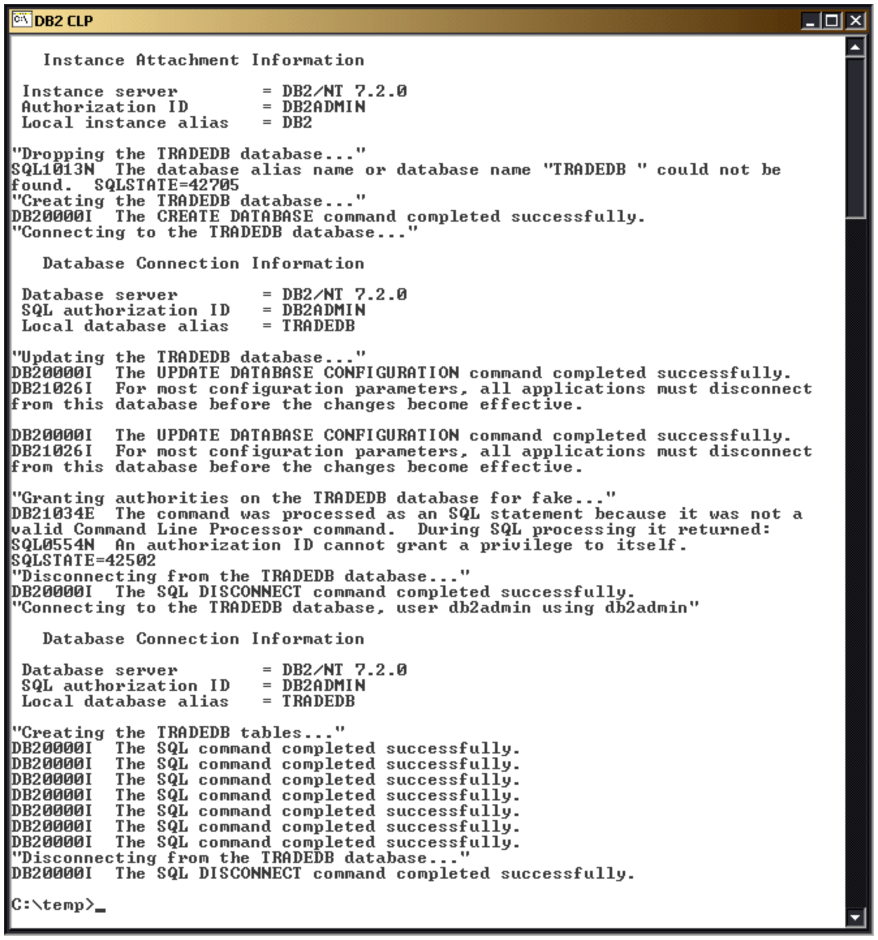
Figure 9-3 Open command window

2. Open a command window and change to the directory where the script resides. Select **Start-> Run** and enter `cmd` in the text field. Click **OK** to start the command window. Refer to Figure 9-3.
3. Change to the directory where the script resides.
4. Issue the following command:

   ```
   db2cmd <script name> <db2user> <db2user password> <instance> <log>
   ```

   For our example, we entered:

   ```
   db2cmd tradeDB.bat db2admin db2admin db2
   ```



DB2 CLP

```
   Instance Attachment Information

 Instance server        = DB2/NT 7.2.0
 Authorization ID       = DB2ADMIN
 Local instance alias   = DB2

"Dropping the TRADEDB database..."
SQL1013N  The database alias name or database name "TRADEDB " could not be
found.  SQLSTATE=42705
"Creating the TRADEDB database..."
DB20000I  The CREATE DATABASE command completed successfully.
"Connecting to the TRADEDB database..."

   Database Connection Information

 Database server        = DB2/NT 7.2.0
 SQL authorization ID   = DB2ADMIN
 Local database alias   = TRADEDB

"Updating the TRADEDB database..."
DB20000I  The UPDATE DATABASE CONFIGURATION command completed successfully.
DB21026I  For most configuration parameters, all applications must disconnect
from this database before the changes become effective.

DB20000I  The UPDATE DATABASE CONFIGURATION command completed successfully.
DB21026I  For most configuration parameters, all applications must disconnect
from this database before the changes become effective.

"Granting authorities on the TRADEDB database for fake..."
DB21034E  The command was processed as an SQL statement because it was not a
valid Command Line Processor command.  During SQL processing it returned:
SQL0554N  An authorization ID cannot grant a privilege to itself.
SQLSTATE=42502
"Disconnecting from the TRADEDB database..."
DB20000I  The SQL DISCONNECT command completed successfully.
"Connecting to the TRADEDB database, user db2admin using db2admin"

   Database Connection Information

 Database server        = DB2/NT 7.2.0
 SQL authorization ID   = DB2ADMIN
 Local database alias   = TRADEDB

"Creating the TRADEDB tables..."
DB20000I  The SQL command completed successfully.
DB20000I  The SQL command completed successfully.
DB20000I  The SQL command completed successfully.
DB20000I  The SQL command completed successfully.
DB20000I  The SQL command completed successfully.
DB20000I  The SQL command completed successfully.
DB20000I  The SQL command completed successfully.
"Disconnecting from the TRADEDB database..."
DB20000I  The SQL DISCONNECT command completed successfully.

C:\temp>_
```

Figure 9-4   Starting the tradeDB script

5. This starts the creation of the database and the tables in it. The output is shown in Figure 9-4.

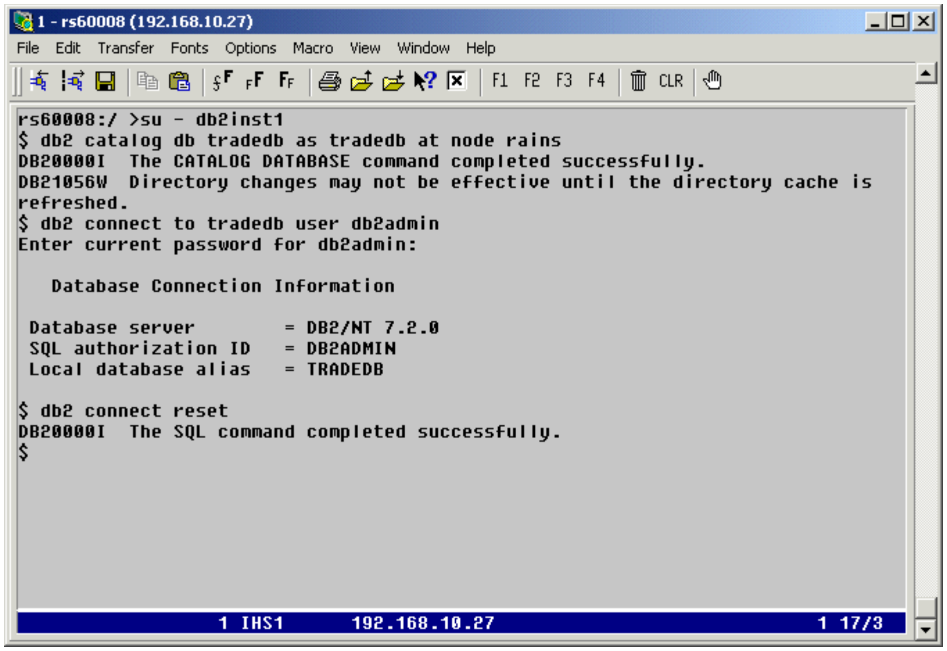
9.2.3 DB clients

Catalog the databases

For each DB client, you must catalog the new databases to be able to access them.

1. On the WebSphere Application Server nodes, log on as the local instance owner.

   ```
   #> su - db2inst1
   ```


1 - rs60008 (192.168.10.27)
File Edit Transfer Fonts Options Macro View Window Help

```
rs60008:/ >su - db2inst1
$ db2 catalog db tradedb as tradedb at node rains
DB20000I  The CATALOG DATABASE command completed successfully.
DB21056W  Directory changes may not be effective until the directory cache is
refreshed.
$ db2 connect to tradedb user db2admin
Enter current password for db2admin:

   Database Connection Information

 Database server        = DB2/NT 7.2.0
 SQL authorization ID   = DB2ADMIN
 Local database alias   = TRADEDB

$ db2 connect reset
DB20000I  The SQL command completed successfully.
$
```

```
                  1 IHS1        192.168.10.27                          1 17/3
```

Figure 9-5 Catalog the databases

2. Issue the catalog command for the database. Figure 12-11 shows the commands for our environment.

```
db2 catalog db <db name> as <db alias> at node <remote Node Name>
```

3. Repeat these steps on each WebSphere Application Server node.

9.2.4 Application servers

Add the new members

To add new members to our existing WebSphere domain without destroying existing configurations, you will have to edit the configuration file for the admin server. It it located in the /bin directory of the WebSphere root directory.

Make sure the to following lines are set as shown:

```
install.initial.config=true
com.ibm.ejs.sm.adminServer.createTables=false
```

This will install an initial configuration without recreating all tables. So you will have the basic resources such as the JDBC driver installed on the node. It will also create a Default Server with the samples application on each node plus a virtual host named default_host. Simply remove these items since you do not need them.

After editing the file, start the admin server. When it shows the message `Open for e-business` in the tracefile, you can start the administrative console.

Create the virtual hosts

1. Open the WebSphere Administrative Console.

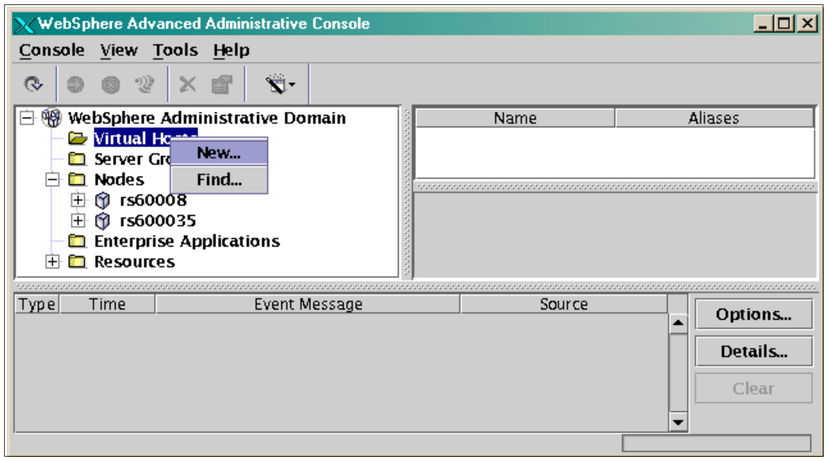
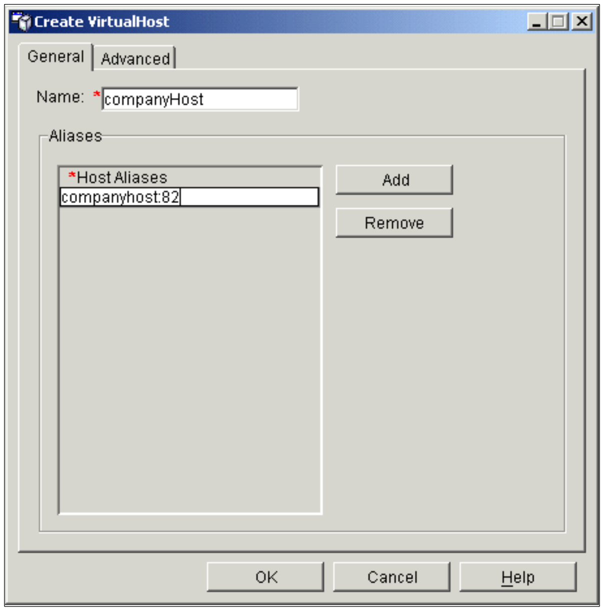


Figure 9-6 Create a new virtual host

2. Right-click **VirtualHosts** in the left-hand pane and select **New**.



Name: * companyHost

Aliases

*Host Aliases

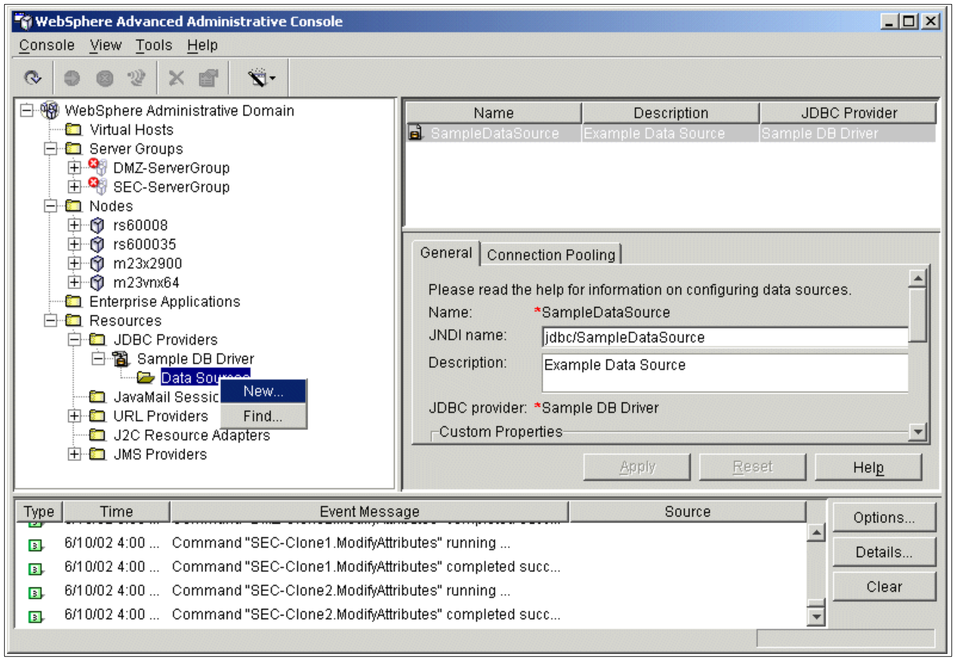
companyhost:82

Figure 9-7   New virtual host name

3. Specify the new hostname. We chose companyHost. Enter a host alias. We chose companyhost:82 because the Web server listens on port 82.
4. Click **OK** to save the changes.

Create the Data Source

1. In the left-hand pane of the WebSphere Administrative Console, open: **Resources -> JDBC Providers -> Sample DB Driver -> Data Sources**.



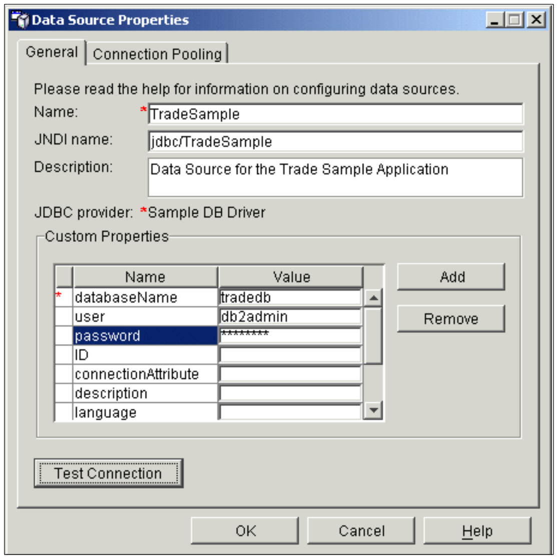
Name	Description	JDBC Provider
SampleDataSource	Example Data Source	Sample DB Driver

Please read the help for information on configuring data sources.
Name: *SampleDataSource
JNDI name: jdbc/SampleDataSource
Description: Example Data Source
JDBC provider: *Sample DB Driver
Custom Properties

Type	Time	Event Message	Source
	6/10/02 4:00 ...	Command "SEC-Clone1.ModifyAttributes" running ...	
	6/10/02 4:00 ...	Command "SEC-Clone1.ModifyAttributes" completed succ...	
	6/10/02 4:00 ...	Command "SEC-Clone2.ModifyAttributes" running ...	
	6/10/02 4:00 ...	Command "SEC-Clone2.ModifyAttributes" completed succ...	

Figure 9-8 Create new Data Source

2. Right-click the **DataSources** item and select **New**.

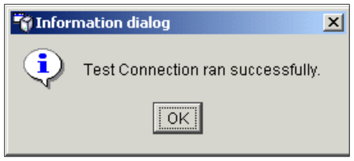


Please read the help for information on configuring data sources.
Name: *TradeSample
JNDI name: jdbc/TradeSample
Description: Data Source for the Trade Sample Application
JDBC provider: *Sample DB Driver
Custom Properties

Name	Value
* databaseName	tradedb
user	db2admin
password	********
ID	
connectionAttribute	
description	
language	

Figure 9-9 Data Source properties

3. Specify the following properties:
   – Name: `TradeSample`
   – JNDI Name: `jdbc/TradeSample`
   – database Name: `tradedb`
   – user: `db2admin`
   – password: db2admin password
4. Click **Test the connection**.


Information dialog

Test Connection ran successfully.

Figure 9-10 Testing the new connection

Create the ServerGroup

1. Go to the WebSphere Administrative Console.

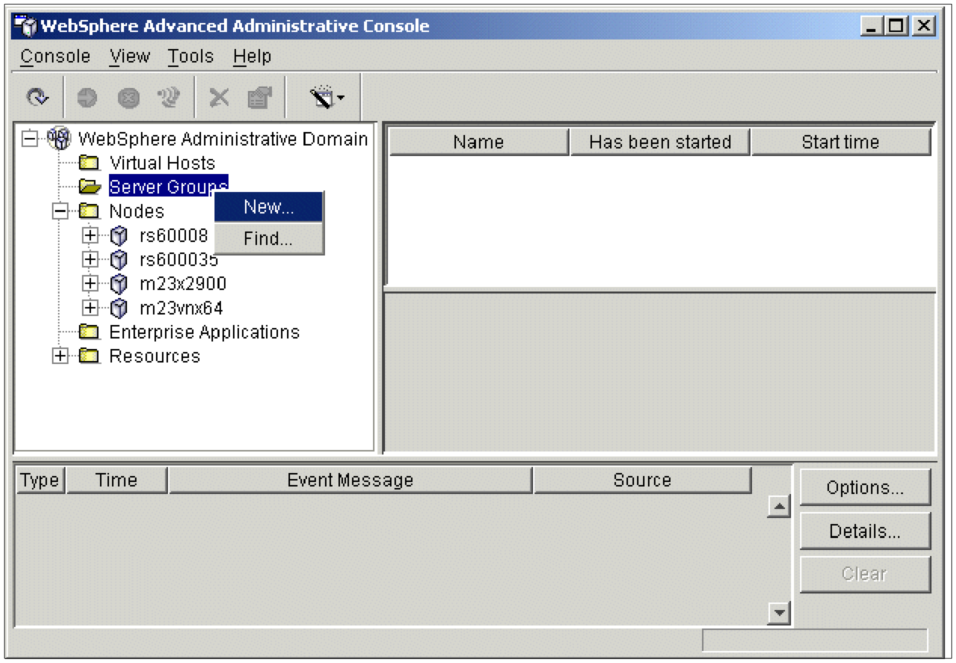
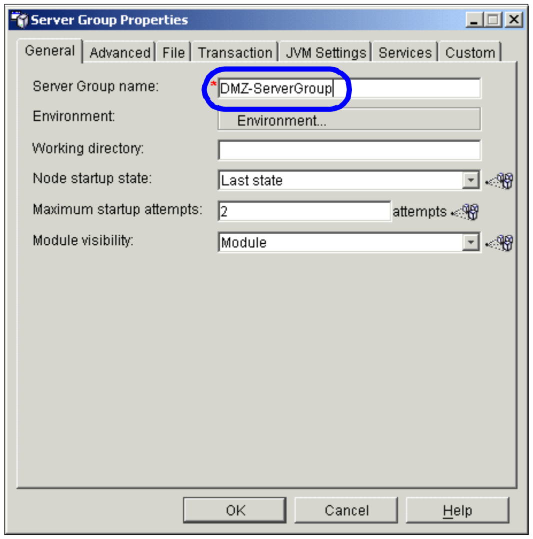

WebSphere Advanced Administrative Console

Figure 9-11 Create a new ServerGroup

2. Right-click **ServerGroups** and select **New** as shown in Figure 9-11.

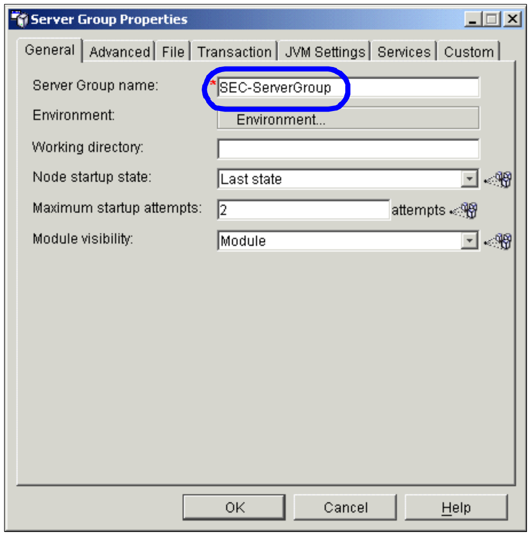

Server Group Properties

General | Advanced | File | Transaction | JVM Settings | Services | Custom

Server Group name:	*DMZ-ServerGroup
Environment:	Environment...
Working directory:	
Node startup state:	Last state
Maximum startup attempts:	2 attempts
Module visibility:	Module

Figure 9-12   Create DMZ ServerGroup

3. Specify `DMZ-ServerGroup` as the name.
4. Create a second ServerGroup.


Server Group Properties

General | Advanced | File | Transaction | JVM Settings | Services | Custom

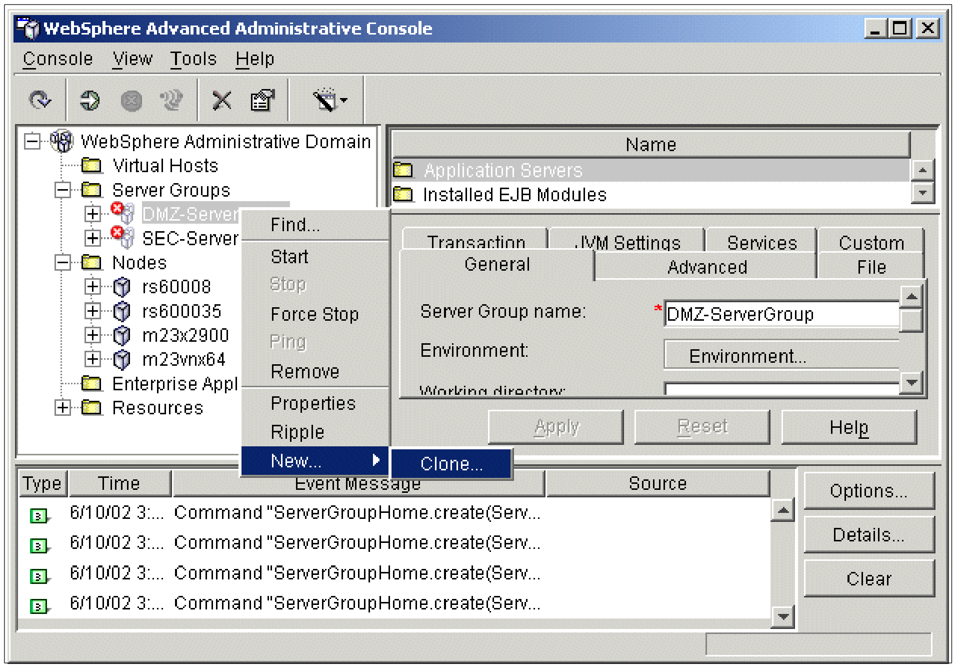
Server Group name:	*SEC-ServerGroup
Environment:	Environment...
Working directory:	
Node startup state:	Last state
Maximum startup attempts:	2 attempts
Module visibility:	Module

Figure 9-13   Create SEC ServerGroup

5. The name of the second ServerGroup is SEC-ServerGroup.

Create the clones

1. In the Administrative Console, right-click a ServerGroup you created.

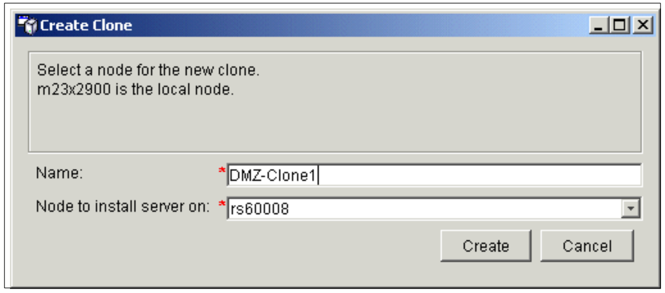


Server Group name: * DMZ-ServerGroup
Environment: Environment...
Working directory:

Type	Time	Event Message	Source
	6/10/02 3:...	Command "ServerGroupHome.create(Serv...	
	6/10/02 3:...	Command "ServerGroupHome.create(Serv...	
	6/10/02 3:...	Command "ServerGroupHome.create(Serv...	
	6/10/02 3:...	Command "ServerGroupHome.create(Serv...	

Figure 9-14 Create new clone

2. Select **New -> Clone**.

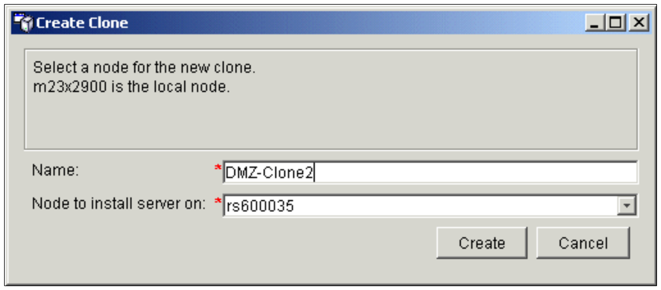


Select a node for the new clone.
m23x2900 is the local node.

Name: * DMZ-Clone1
Node to install server on: * rs60008

Figure 9-15 Properties of DMZ-Clone1

3. Create two clones for each ServerGroup: one on each AppServer in the DMZ for the DMZ-ServerGroup, and one on each AppServer in the secure network for the SEC-ServerGroup.

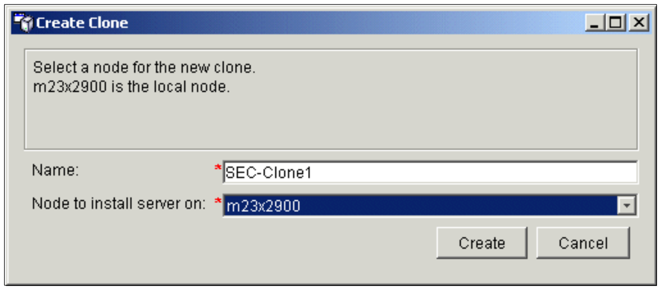

Create Clone

Select a node for the new clone.
m23x2900 is the local node.

Name: * DMZ-Clone2

Node to install server on: * rs600035

Figure 9-16 Properties of DMZ-Clone2

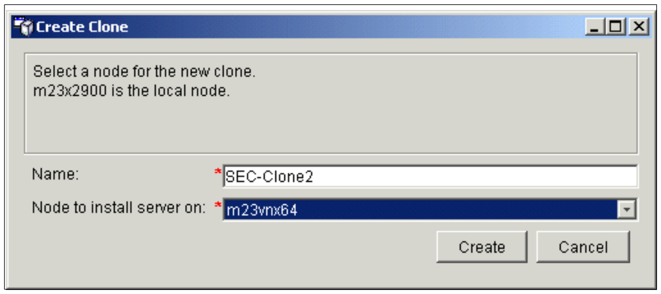

Create Clone

Select a node for the new clone.
m23x2900 is the local node.

Name: * SEC-Clone1

Node to install server on: * m23x2900

Figure 9-17 Properties of SEC-Clone1

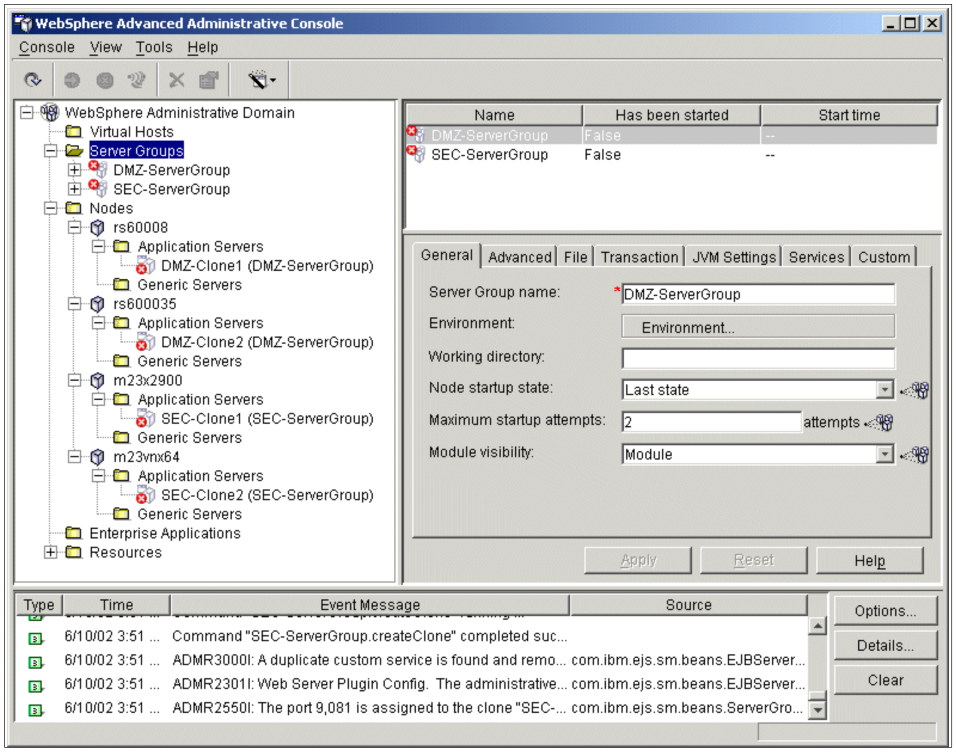

Create Clone

Select a node for the new clone.
m23x2900 is the local node.

Name: * SEC-Clone2

Node to install server on: * m23vnx64

Figure 9-18 Properties of SEC-Clone2



Name	Has been started	Start time
DMZ-ServerGroup	False	--
SEC-ServerGroup	False	--

Server Group name: DMZ-ServerGroup
Environment: Environment...
Working directory:
Node startup state: Last state
Maximum startup attempts: 2 attempts
Module visibility: Module

Type	Time	Event Message	Source
	6/10/02 3:51 ...	Command "SEC-ServerGroup.createClone" completed suc...	
	6/10/02 3:51 ...	ADMR3000I: A duplicate custom service is found and remo...	com.ibm.ejs.sm.beans.EJBServer...
	6/10/02 3:51 ...	ADMR2301I: Web Server Plugin Config. The administrative...	com.ibm.ejs.sm.beans.EJBServer...
	6/10/02 3:51 ...	ADMR2550I: The port 9,081 is assigned to the clone "SEC-...	com.ibm.ejs.sm.beans.ServerGro...

Figure 9-19 All ServerGroups and clones

Figure 9-19 illustrates the WebSphere Application Server console after all server groups and clones have been defined.

Configure the clones

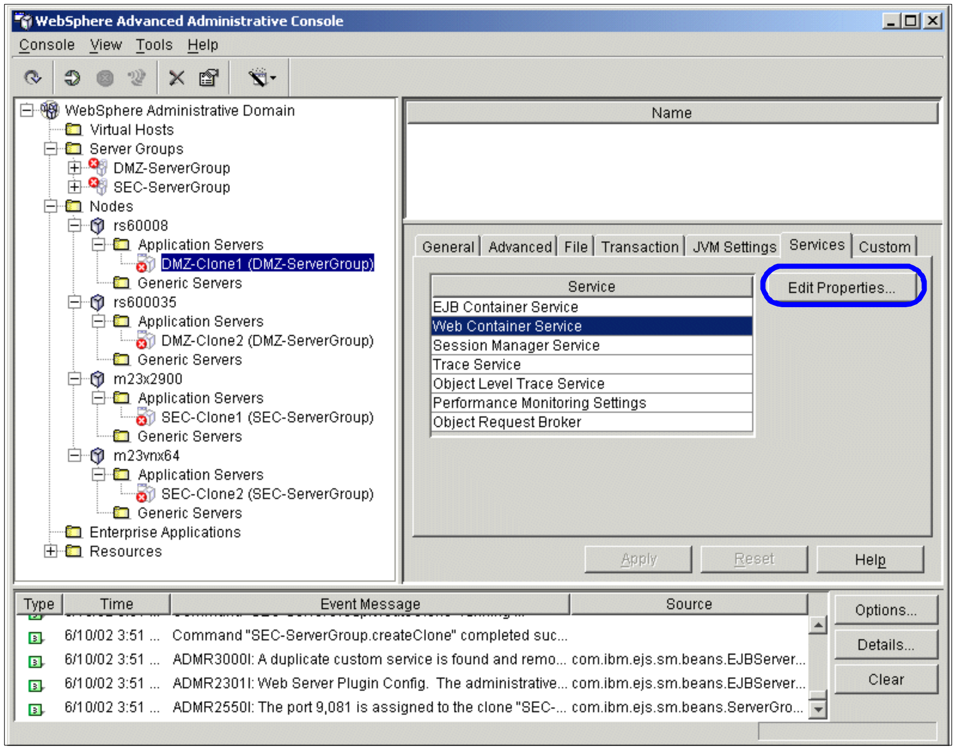
We now configure the clones to use the correct hostnames and distinct ports.

Table 9-3 Ports used by the clones

Node	ServerGroup	Clone	Port
RS60008	DMZ-ServerGroup	DMZ-Clone1	8301
RS600035	DMZ-ServerGroup	DMZ-Clone2	8302
M23X2900	SEC-ServerGroup	SEC-Clone1	8311
M23VNX64	SEC-ServerGroup	SEC-Clone2	8312

1. Again, we will configure the clones to use distinct port numbers. Select a clone in the left-hand pane of the Administrative Console.

2. In the right-hand pane, select **Web Container Service**. See Figure 9-20.

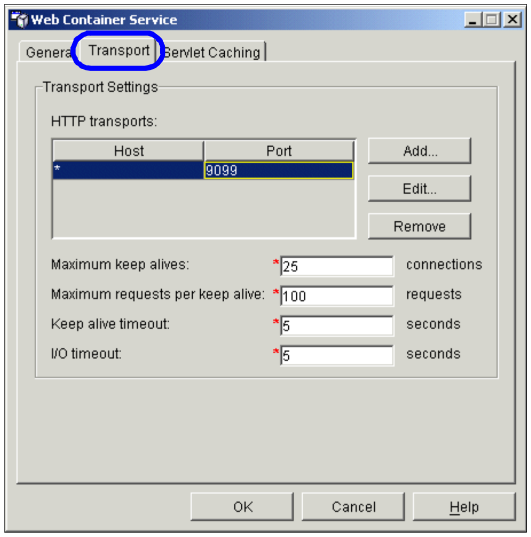


Service
EJB Container Service
Web Container Service
Session Manager Service
Trace Service
Object Level Trace Service
Performance Monitoring Settings
Object Request Broker

Type	Time	Event Message	Source
	6/10/02 3:51 ...	Command "SEC-ServerGroup.createClone" completed suc...	
	6/10/02 3:51 ...	ADMR3000I: A duplicate custom service is found and remo...	com.ibm.ejs.sm.beans.EJBServer...
	6/10/02 3:51 ...	ADMR2301I: Web Server Plugin Config. The administrative...	com.ibm.ejs.sm.beans.EJBServer...
	6/10/02 3:51 ...	ADMR2550I: The port 9,081 is assigned to the clone "SEC-...	com.ibm.ejs.sm.beans.ServerGro...

Figure 9-20 Configure the clones

3. Click **Edit Properties**.


Web Container Service

General | Transport | Servlet Caching

Transport Settings

HTTP transports:

Host	Port
*	9099

Maximum keep alives: *25 connections

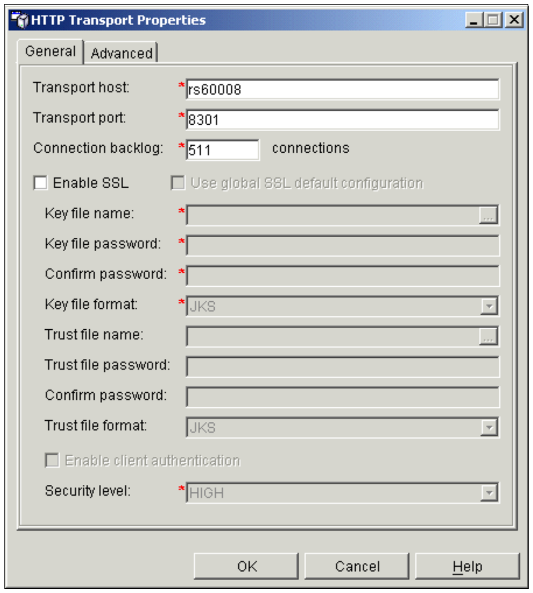
Maximum requests per keep alive: *100 requests

Keep alive timeout: *5 seconds

I/O timeout: *5 seconds

Figure 9-21 Clone properties

4. Select the **Transport** pane.
5. Select the entry in the Host/Port list.
6. Click **Edit**.


HTTP Transport Properties

General | Advanced

Transport host: *rs60008

Transport port: *8301

Connection backlog: *511 connections

☐ Enable SSL ☐ Use global SSL default configuration

Key file name: *

Key file password: *

Confirm password: *

Key file format: *JKS

Trust file name:

Trust file password:

Confirm password:

Trust file format: JKS

☐ Enable client authentication

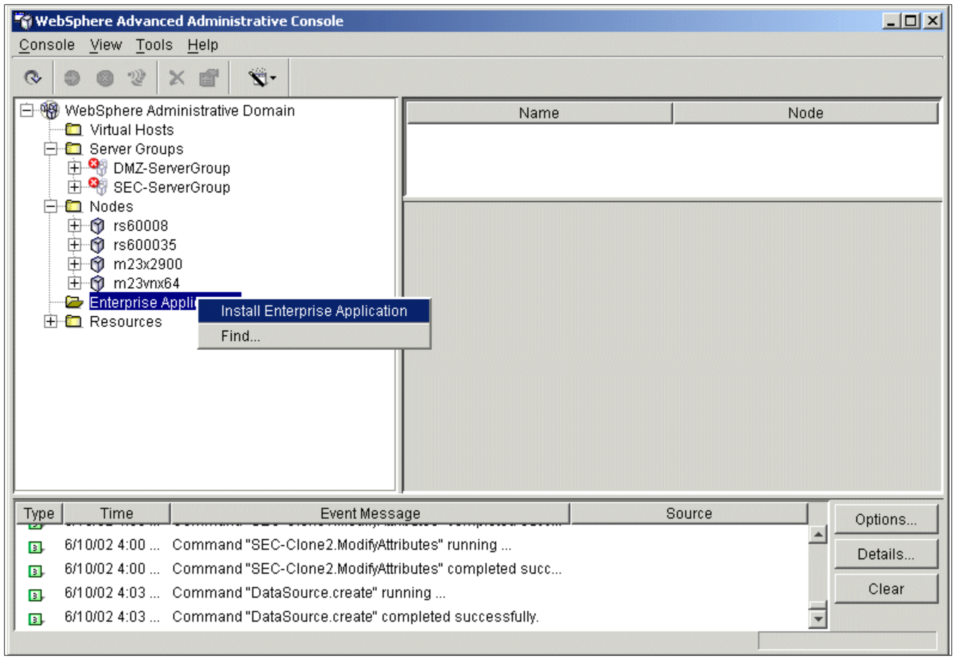
Security level: *HIGH

Figure 9-22 Transport properties

7. Specify the Transport Host and Transport Port properties and click **OK**.
8. Repeat these steps with each clone. The ports are shown in Table 9-3, “Ports used by the clones” on page 317.

Install the Trade application

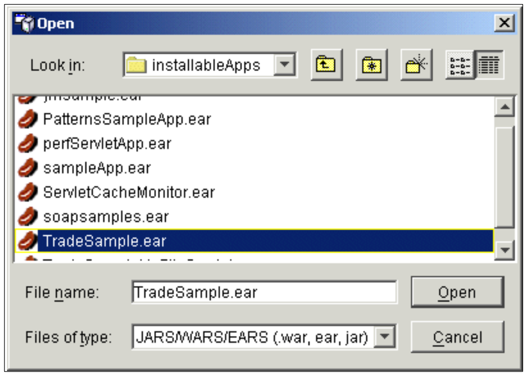
1. Go to the WebSphere Administrative Console.



```
6/10/02 4:00 ...  Command "SEC-Clone2.ModifyAttributes" running ...
6/10/02 4:00 ...  Command "SEC-Clone2.ModifyAttributes" completed succ...
6/10/02 4:03 ...  Command "DataSource.create" running ...
6/10/02 4:03 ...  Command "DataSource.create" completed successfully.
```

Figure 9-23 Install Enterprise Application

2. In the left-hand pane right-click **Enterprise Application** and select **Install Enterprise Application**.

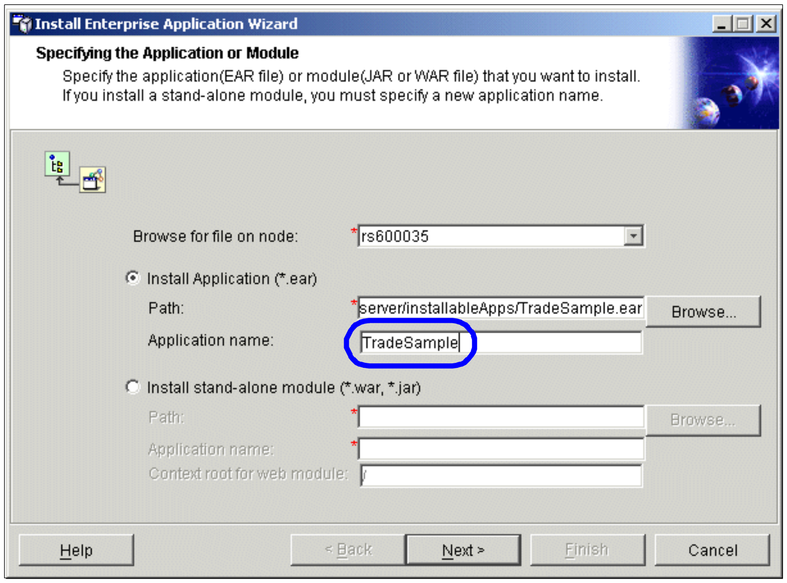


File name: TradeSample.ear

Files of type: JARS/WARS/EARS (.war, ear, jar)

*Figure 9-24 Select application *.ear file*

3. Browse to the installableApps directory and select the **TradeSample.ear** file.



Specifying the Application or Module

Specify the application(EAR file) or module(JAR or WAR file) that you want to install. If you install a stand-alone module, you must specify a new application name.

Browse for file on node: * rs600035

(●) Install Application (*.ear)

Path: * server/installableApps/TradeSample.ear

Application name: TradeSample

( ) Install stand-alone module (*.war, *.jar)

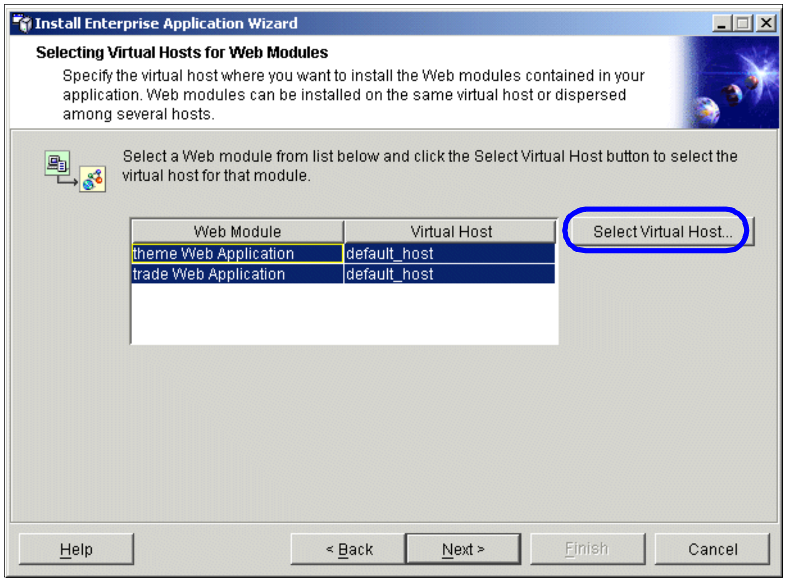
Path: *

Application name: *

Context root for web module: /

Figure 9-25 Installation Wizard

4. Specify a name for the application. We chose TradeSample.
5. Click **Next** until you reach the Virtual Host selection window. See Figure 9-28.


Install Enterprise Application Wizard

Selecting Virtual Hosts for Web Modules

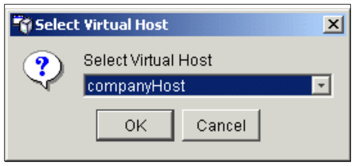
Specify the virtual host where you want to install the Web modules contained in your application. Web modules can be installed on the same virtual host or dispersed among several hosts.

Select a Web module from list below and click the Select Virtual Host button to select the virtual host for that module.

Web Module	Virtual Host
theme Web Application	default_host
trade Web Application	default_host

Figure 9-26 Virtual Host selection

6. For each Web module, click **Select Virtual Host...** to select which virtual host the module will run on.

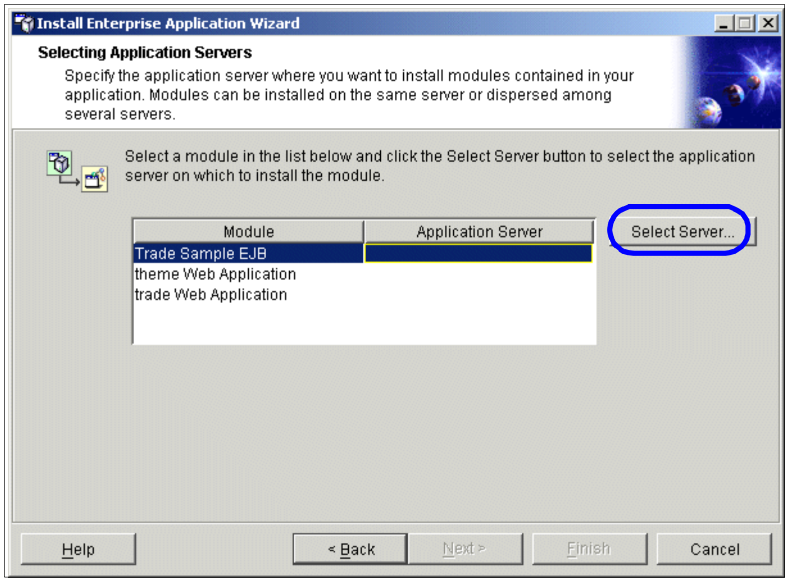

Select Virtual Host

Select Virtual Host

companyHost

Figure 9-27 Select Virtual Host

7. For the two Web modules, select the **companyHost**.
8. Click **Next** until you reach the Application Server selection window.


Install Enterprise Application Wizard

Selecting Application Servers

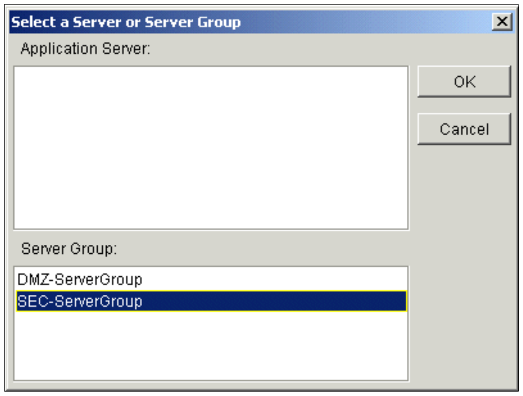
Specify the application server where you want to install modules contained in your application. Modules can be installed on the same server or dispersed among several servers.

Select a module in the list below and click the Select Server button to select the application server on which to install the module.

Module	Application Server
Trade Sample EJB	
theme Web Application	
trade Web Application	

Figure 9-28 Choose Application Server

9. Select the **Trade Sample EJB** module and click **Select Server**.


Select a Server or Server Group

Application Server:

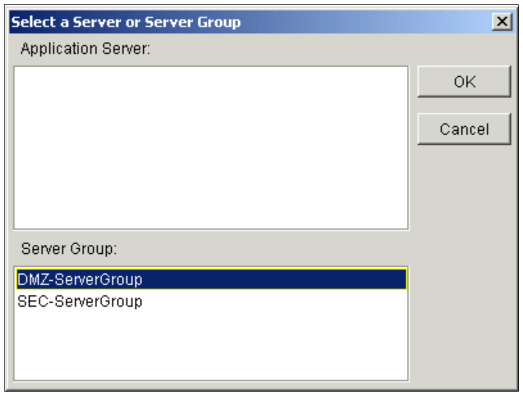
Server Group:

DMZ-ServerGroup
SEC-ServerGroup

Figure 9-29 Select Application Server 1

10. Select the **SEC-ServerGroup** for this module and click **OK** to accept the selection.

11. Select the two Web application modules and click **Select Server**.



Select a Server or Server Group

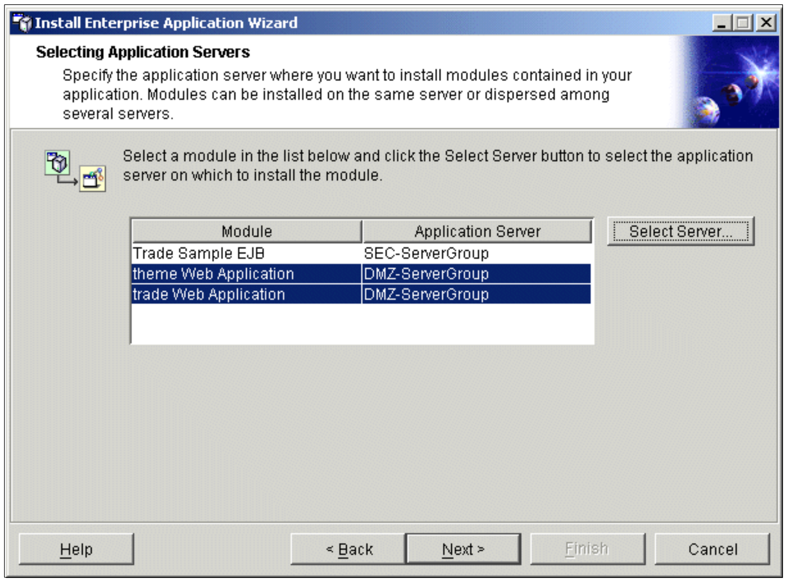
Application Server:

Server Group:

DMZ-ServerGroup
SEC-ServerGroup

Figure 9-30 Select Application Server 2

12. For the Web application modules, select the **DMZ-ServerGroup** and click **OK** to accept the selection.



Install Enterprise Application Wizard

Selecting Application Servers

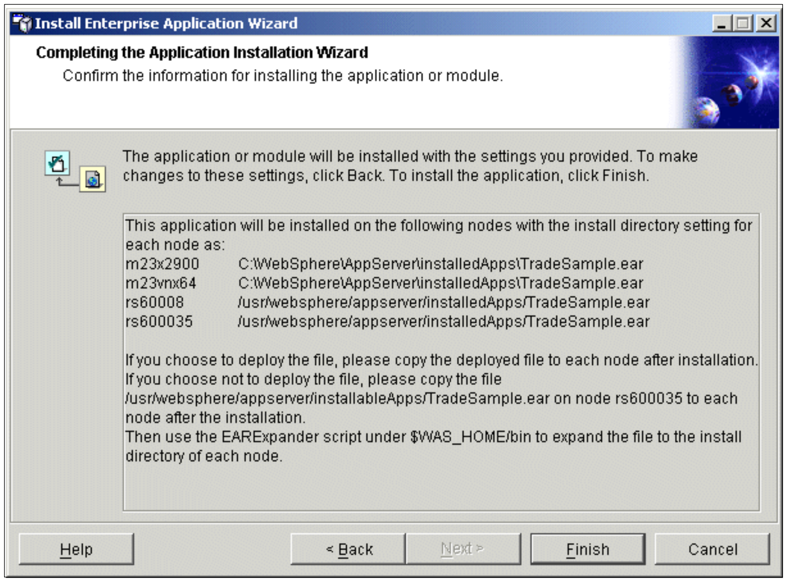
Specify the application server where you want to install modules contained in your application. Modules can be installed on the same server or dispersed among several servers.

Select a module in the list below and click the Select Server button to select the application server on which to install the module.

Module	Application Server
Trade Sample EJB	SEC-ServerGroup
theme Web Application	DMZ-ServerGroup
trade Web Application	DMZ-ServerGroup

Figure 9-31 The selected Application Servers

13. Figure 9-31 shows the selected Application Servers for each module.

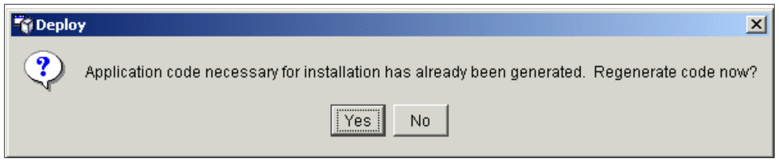

Install Enterprise Application Wizard
Completing the Application Installation Wizard
Confirm the information for installing the application or module.
Help
< Back
Next >
Finish
Cancel

The application or module will be installed with the settings you provided. To make changes to these settings, click Back. To install the application, click Finish.

```
This application will be installed on the following nodes with the install directory setting for
each node as:
m23x2900      C:\WebSphere\AppServer\installedApps\TradeSample.ear
m23vnx64      C:\WebSphere\AppServer\installedApps\TradeSample.ear
rs60008       /usr/websphere/appserver/installedApps/TradeSample.ear
rs600035      /usr/websphere/appserver/installedApps/TradeSample.ear

If you choose to deploy the file, please copy the deployed file to each node after installation.
If you choose not to deploy the file, please copy the file
/usr/websphere/appserver/installableApps/TradeSample.ear on node rs600035 to each
node after the installation.
Then use the EARExpander script under $WAS_HOME/bin to expand the file to the install
directory of each node.
```

Figure 9-32 Start Installation window

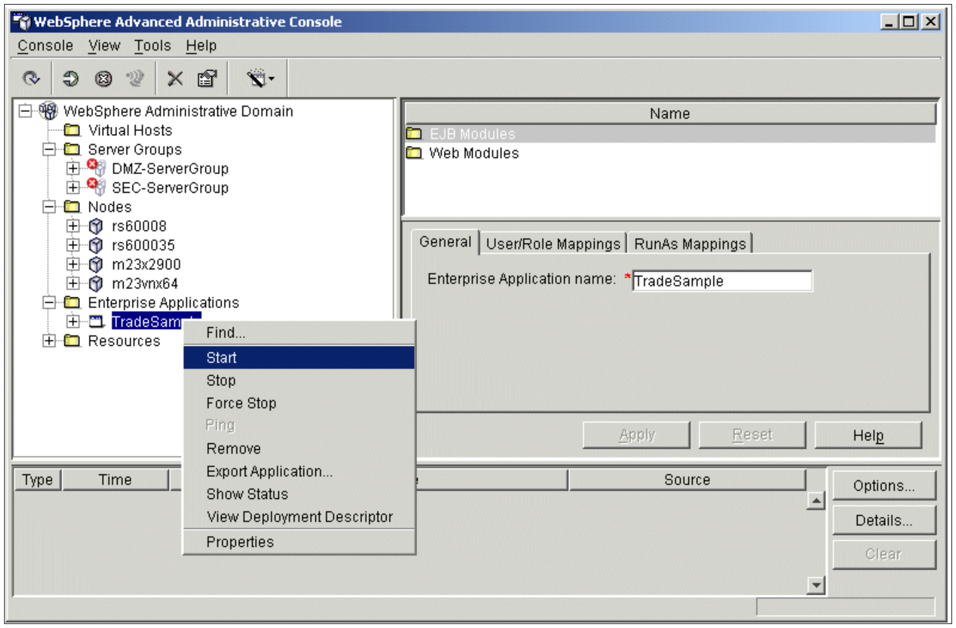
14. Click **Finish** to start the installation process.



Application code necessary for installation has already been generated. Regenerate code now?

Figure 9-33 Code regeneration

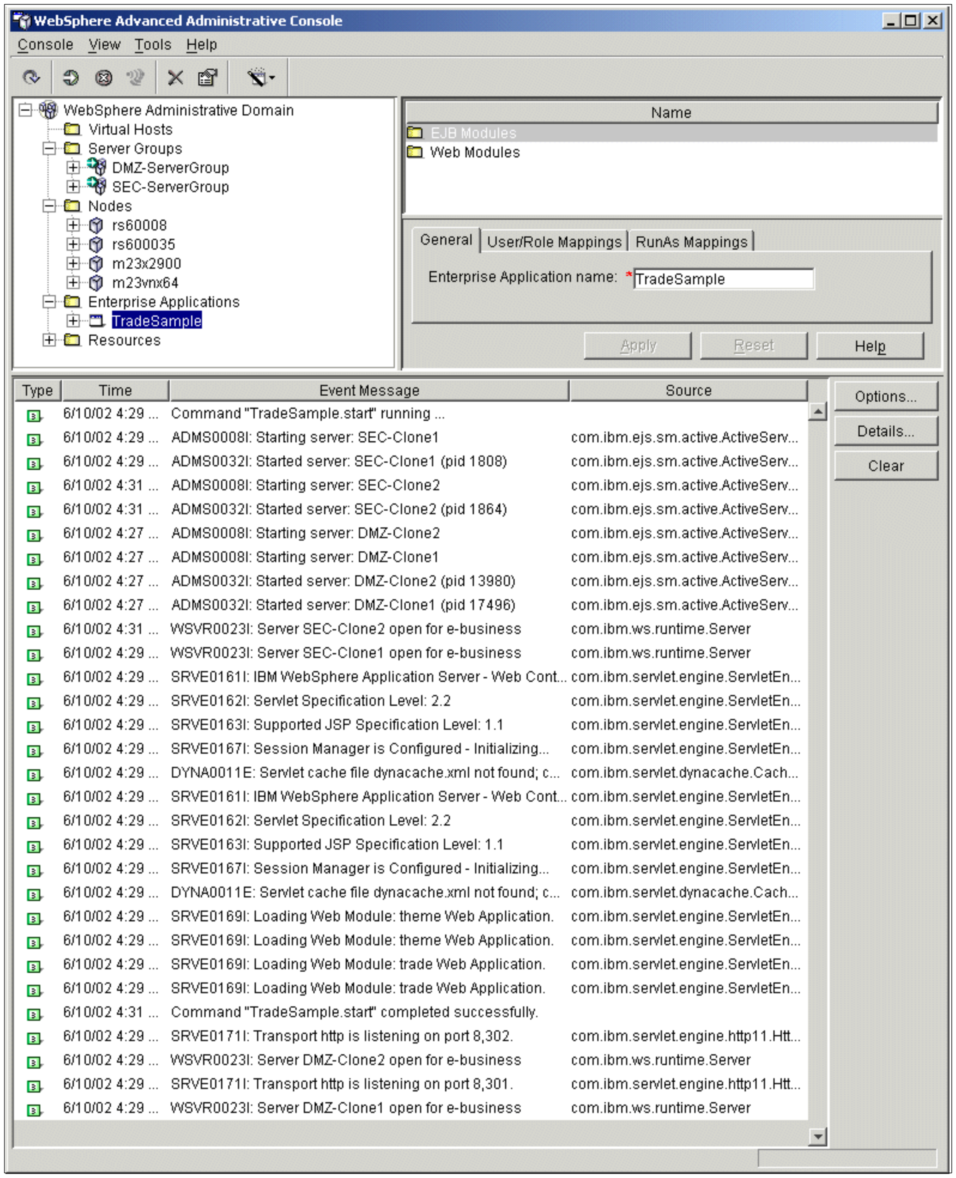
15. A pop-up window gives you the possibility to regenerate the code. Click **No** in this case.



Enterprise Application name: *TradeSample

Figure 9-34 Start Enterprise Application

16. Right-click the **TradeSample** application and select **Start**.



- WebSphere Administrative Domain
  - Virtual Hosts
  - Server Groups
    - DMZ-ServerGroup
    - SEC-ServerGroup
  - Nodes
    - rs60008
    - rs600035
    - m23x2900
    - m23vnx64
  - Enterprise Applications
    - TradeSample
  - Resources

Enterprise Application name: *TradeSample

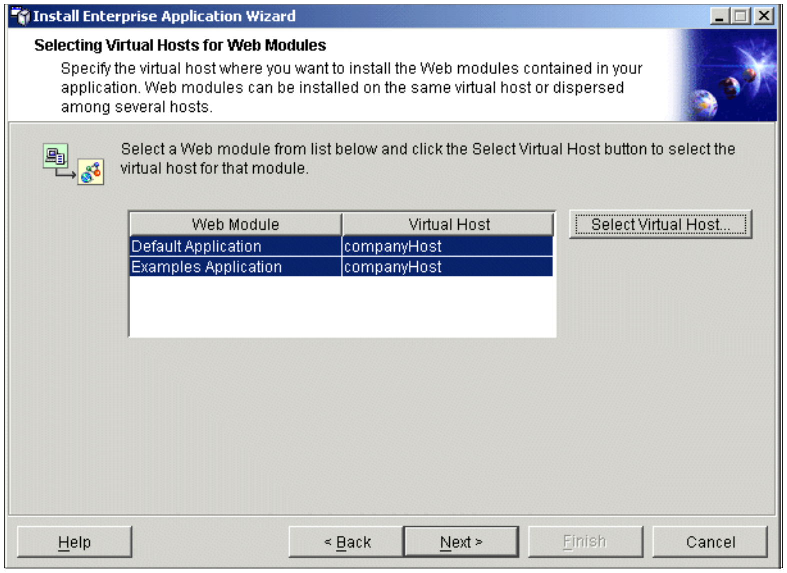
Type	Time	Event Message	Source
	6/10/02 4:29 ...	Command "TradeSample.start" running ...	
	6/10/02 4:29 ...	ADMS0008I: Starting server: SEC-Clone1	com.ibm.ejs.sm.active.ActiveServ...
	6/10/02 4:29 ...	ADMS0032I: Started server: SEC-Clone1 (pid 1808)	com.ibm.ejs.sm.active.ActiveServ...
	6/10/02 4:31 ...	ADMS0008I: Starting server: SEC-Clone2	com.ibm.ejs.sm.active.ActiveServ...
	6/10/02 4:31 ...	ADMS0032I: Started server: SEC-Clone2 (pid 1864)	com.ibm.ejs.sm.active.ActiveServ...
	6/10/02 4:27 ...	ADMS0008I: Starting server: DMZ-Clone2	com.ibm.ejs.sm.active.ActiveServ...
	6/10/02 4:27 ...	ADMS0008I: Starting server: DMZ-Clone1	com.ibm.ejs.sm.active.ActiveServ...
	6/10/02 4:27 ...	ADMS0032I: Started server: DMZ-Clone2 (pid 13980)	com.ibm.ejs.sm.active.ActiveServ...
	6/10/02 4:27 ...	ADMS0032I: Started server: DMZ-Clone1 (pid 17496)	com.ibm.ejs.sm.active.ActiveServ...
	6/10/02 4:31 ...	WSVR0023I: Server SEC-Clone2 open for e-business	com.ibm.ws.runtime.Server
	6/10/02 4:29 ...	WSVR0023I: Server SEC-Clone1 open for e-business	com.ibm.ws.runtime.Server
	6/10/02 4:29 ...	SRVE0161I: IBM WebSphere Application Server - Web Cont...	com.ibm.servlet.engine.ServletEn...
	6/10/02 4:29 ...	SRVE0162I: Servlet Specification Level: 2.2	com.ibm.servlet.engine.ServletEn...
	6/10/02 4:29 ...	SRVE0163I: Supported JSP Specification Level: 1.1	com.ibm.servlet.engine.ServletEn...
	6/10/02 4:29 ...	SRVE0167I: Session Manager is Configured - Initializing...	com.ibm.servlet.engine.ServletEn...
	6/10/02 4:29 ...	DYNA0011E: Servlet cache file dynacache.xml not found; c...	com.ibm.servlet.dynacache.Cach...
	6/10/02 4:29 ...	SRVE0161I: IBM WebSphere Application Server - Web Cont...	com.ibm.servlet.engine.ServletEn...
	6/10/02 4:29 ...	SRVE0162I: Servlet Specification Level: 2.2	com.ibm.servlet.engine.ServletEn...
	6/10/02 4:29 ...	SRVE0163I: Supported JSP Specification Level: 1.1	com.ibm.servlet.engine.ServletEn...
	6/10/02 4:29 ...	SRVE0167I: Session Manager is Configured - Initializing...	com.ibm.servlet.engine.ServletEn...
	6/10/02 4:29 ...	DYNA0011E: Servlet cache file dynacache.xml not found; c...	com.ibm.servlet.dynacache.Cach...
	6/10/02 4:29 ...	SRVE0169I: Loading Web Module: theme Web Application.	com.ibm.servlet.engine.ServletEn...
	6/10/02 4:29 ...	SRVE0169I: Loading Web Module: theme Web Application.	com.ibm.servlet.engine.ServletEn...
	6/10/02 4:29 ...	SRVE0169I: Loading Web Module: trade Web Application.	com.ibm.servlet.engine.ServletEn...
	6/10/02 4:29 ...	SRVE0169I: Loading Web Module: trade Web Application.	com.ibm.servlet.engine.ServletEn...
	6/10/02 4:31 ...	Command "TradeSample.start" completed successfully.	
	6/10/02 4:29 ...	SRVE0171I: Transport http is listening on port 8,302.	com.ibm.servlet.engine.http11.Htt...
	6/10/02 4:29 ...	WSVR0023I: Server DMZ-Clone2 open for e-business	com.ibm.ws.runtime.Server
	6/10/02 4:29 ...	SRVE0171I: Transport http is listening on port 8,301.	com.ibm.servlet.engine.http11.Htt...
	6/10/02 4:29 ...	WSVR0023I: Server DMZ-Clone1 open for e-business	com.ibm.ws.runtime.Server

Figure 9-35 Messages on the ServerGroup start

17. This will start all Application Servers that serve this application and the application itself.

Install sample application

1. From the WebSphere Application Server Administrative Console, right-click **Enterprise Application** and select **Install Enterprise Application**.
2. Browse for the SampleApp.ear file.
3. Accept the defaults and proceed until you reach the virtual hosts window.


Install Enterprise Application Wizard

Selecting Virtual Hosts for Web Modules

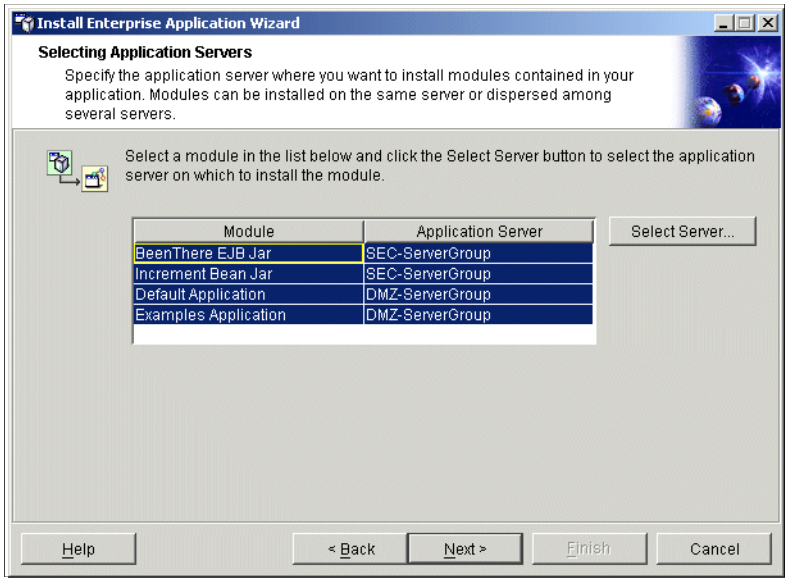
Specify the virtual host where you want to install the Web modules contained in your application. Web modules can be installed on the same virtual host or dispersed among several hosts.

Select a Web module from list below and click the Select Virtual Host button to select the virtual host for that module.

Web Module	Virtual Host
Default Application	companyHost
Examples Application	companyHost

Figure 9-36 Select virtual hosts

4. Select both Web modules and choose the **companyHost** as the virtual host for both modules.


Install Enterprise Application Wizard
Selecting Application Servers

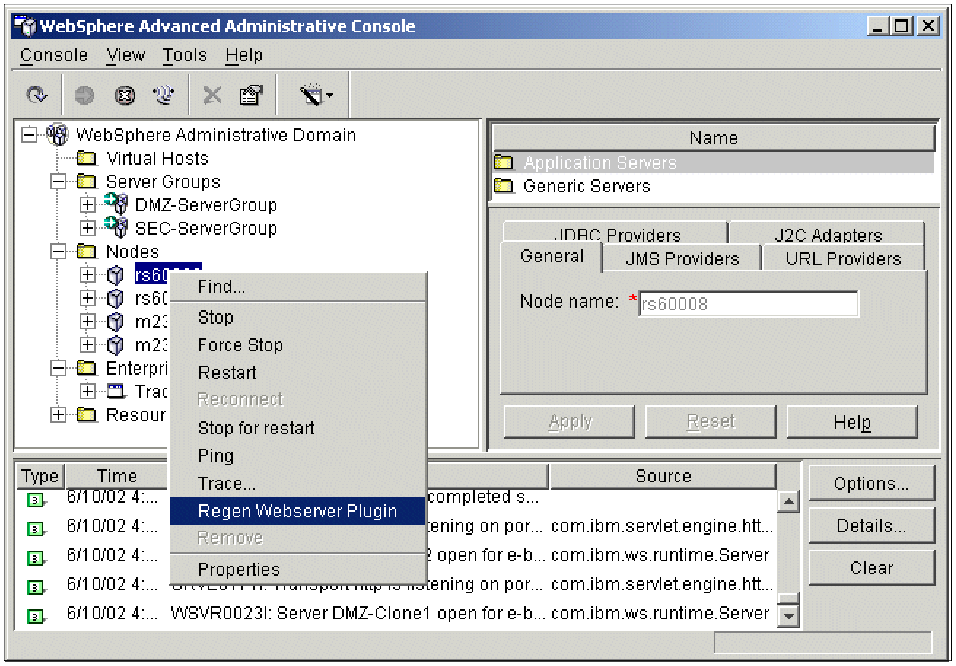
Specify the application server where you want to install modules contained in your application. Modules can be installed on the same server or dispersed among several servers.

Select a module in the list below and click the Select Server button to select the application server on which to install the module.

Module	Application Server
BeenThere EJB Jar	SEC-ServerGroup
Increment Bean Jar	SEC-ServerGroup
Default Application	DMZ-ServerGroup
Examples Application	DMZ-ServerGroup

Figure 9-37 Select Application Server

5. Select the Web application modules and choose the **DMZ-ServerGroup** as the Application Server.
6. Select the EJB modules and choose the **SEC-ServerGroup** as the Application Server.
7. Complete the rest of the installation using the default settings.



Node name: * rs60008

```
6/10/02 4:...                  completed s...
6/10/02 4:...                  tening on por...  com.ibm.servlet.engine.htt...
6/10/02 4:...                  2 open for e-b... com.ibm.ws.runtime.Server
6/10/02 4:...                  tening on por...  com.ibm.servlet.engine.htt...
6/10/02 4:... WSVR0023I: Server DMZ-Clone1 open for e-b... com.ibm.ws.runtime.Server
```

Figure 9-38 Regenerate the Web Server plug-in

8. Do not forget to regenerate the plug-in configuration. Right-click each node and select **Regen Webserver Plugin**.

9.2.5 HTTP servers

Virtual hosts

As long as all content is delivered by WebSphere Application Server through the file servlet, you will not need to define any virtual hosts in the HTTP server's httpd.conf configuration file.

Configure listener

Open the httpd.conf file and add a LISTEN directive for port 82.

```
Listen 82
```

Restart the Web server.

Alias the loopback adapter

As in previous examples, we use the MAC forwarding method for load balancing. In order to accept IP packets for the load balancer cluster address

192.168.10.90, the HTTP server needs to own the IP address. To prevent the server from answering ARP broadcasts, we alias it to the loopback interface. As our HTTP server is on a UNIX machine, we added the alias with the **ifconfig** command:

```
ifconfig lo0 alias 192.168.10.90
```

9.2.6 Caching Proxies

Alias the loopback adapter

Referring to Table 9-1 on page 304, recall that the load balancer cluster IP address for the caching proxy machines represents the company IP address to the Internet. Because we use the MAC forwarding method for load balancing, this load balancer cluster address (192.168.10.80) must be aliased to the loopback interface for each of the caching proxy machines. Since our caching proxy machines are both Windows 2000 machines, we add the load balancer cluster address to the loopback adapter using the Windows GUI. For details on this task, see 6.2.1, “Aliasing a loopback adapter” on page 165.

Proxy Server configuration

We configured the caching proxy machines by editing the ibmproxy.conf file manually on each machine.

Basic configuration

- `ServerRoot C:\Progra~1\IBM\edge\cp\bin`
- `Port 80`
- `AdminPort 8008`

Mapping rules

To enable proxying to the content servers, we need to add a mapping rule:

```
proxy /* http://companyhost:82/* www.company.com
```

Note that the port number of 82 in the above mapping rule must match the LISTEN address configured in the HTTP server. See “Configure listener” on page 330.

Cache behavior

To configure the caching proxy machine for memory caching, we edited the ibmproxy.conf file. The relevant entries are:

- `Cache ON`
- `#CacheDev \\.\D:` (Make sure this is commented out for memory caching)
- `CacheMemory 64 M`

All other settings where left to the default.

9.2.7 Load balancers

Follow the installation guide 6.1.2, “Load Balancer” on page 132 to install the load balancer software on your dispatcher servers.

The mutual high availability load balancer nodes have two clusters configured:

- 192.168.10.80 - www.company (caching proxy load balancer cluster)
- 192.168.10.90 - companyhost (Application Server load balancer cluster)

The first load balancer (192.168.10.14) balances the incoming requests between the caching proxy nodes (cluster 192.168.10.80). The second load balancer (192.168.10.15) receives the forwarded requests from the caching proxy nodes and balances them to the content servers (cluster 192.168.10.90). Remember that in a mutual high availability configuration, each load balancer has its own cluster(s) defined that it load balances. In addition, it also has clusters defined for which it can perform backup load balancing functions if the primary load balancer for that cluster fails. Refer to 4.2.6, “Load Balancer Mutual High Availability feature” on page 73 for details.

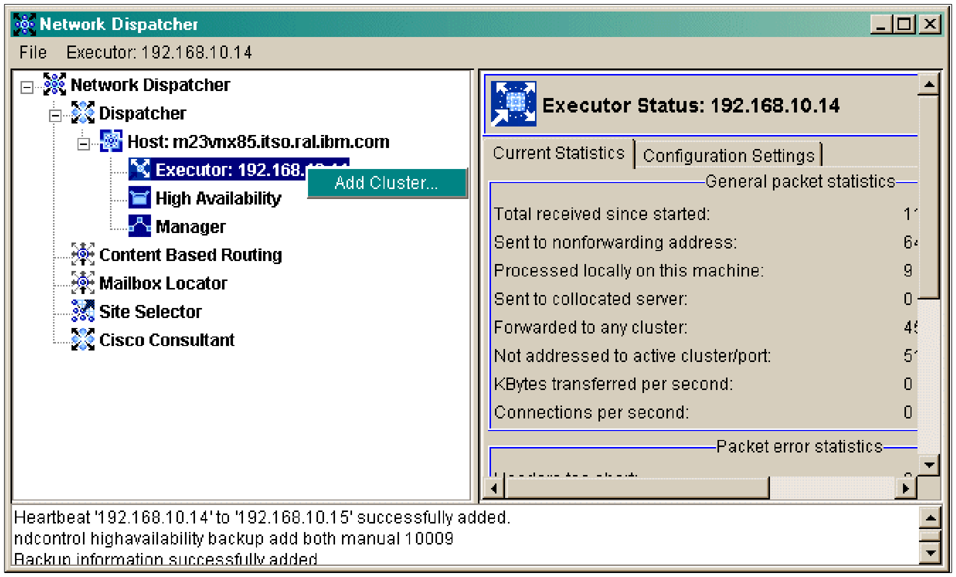
Each load balancer in a mutual high availability configuration must be configured with the same sets of clusters, servers, ports, and so on.

Start the Admin GUI

In Windows, select: **Start -> Programs -> WebSphere -> Edge Server -> IBM Network Dispatcher**. Connect to the host by right-clicking the **Dispatcher** item and selecting **Connect to Host**.

Create the caching proxy clusters

First, we will create a cluster for the caching proxy servers.

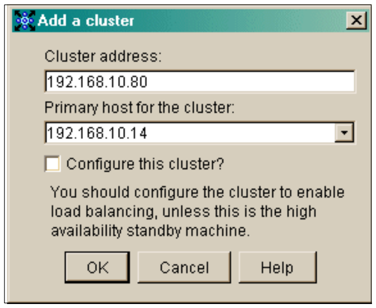


Total received since started: 1
Sent to nonforwarding address: 6
Processed locally on this machine: 9
Sent to collocated server: 0
Forwarded to any cluster: 4
Not addressed to active cluster/port: 5
KBytes transferred per second: 0
Connections per second: 0

```
Heartbeat '192.168.10.14' to '192.168.10.15' successfully added.
ndcontrol highavailability backup add both manual 10009
Backup information successfully added
```

Figure 9-39 Add a cluster

1. Right-click the **Executor** item in the left-hand pane and select **Add Cluster**.



Cluster address: 192.168.10.80

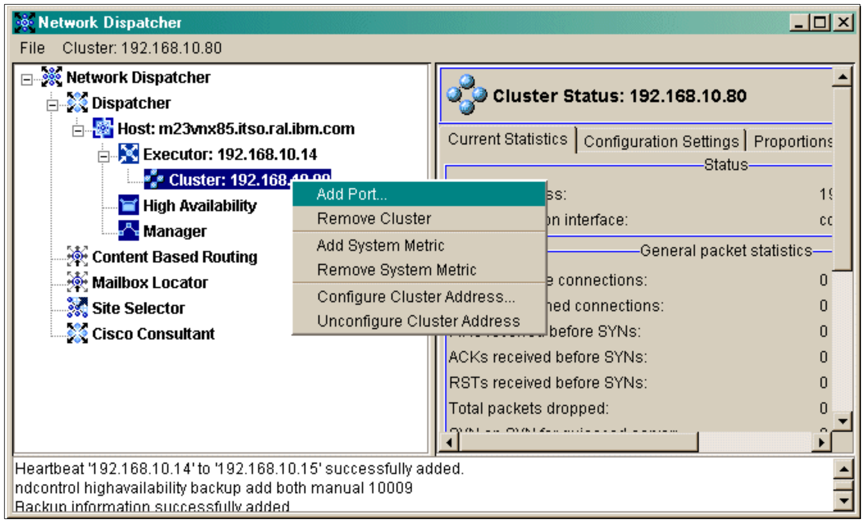
Primary host for the cluster: 192.168.10.14

☐ Configure this cluster?

You should configure the cluster to enable load balancing, unless this is the high availability standby machine.

Figure 9-40 Cluster properties

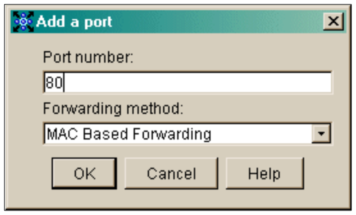
2. Specify the cluster address. In our example it is 192.168.10.80. Do not select the **Configure this cluster?** box.



```
Heartbeat '192.168.10.14' to '192.168.10.15' successfully added.
ndcontrol highavailability backup add both manual 10009
Backup information successfully added
```

Figure 9-41 Add port

3. Now right-click the **Cluster** item in the left-hand pane and select **Add Port**.

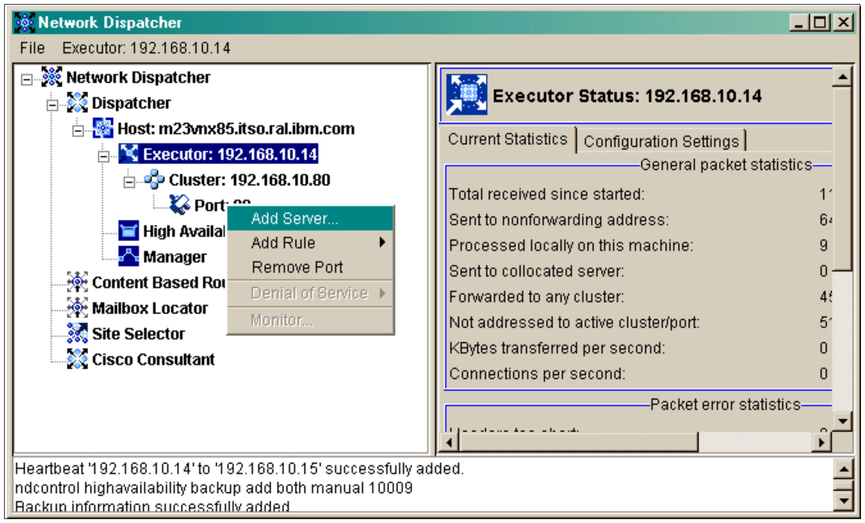


Port number: 80

Forwarding method: MAC Based Forwarding

Figure 9-42 Add port

4. Because our proxy servers will be accessed from the Internet, we chose port 80.

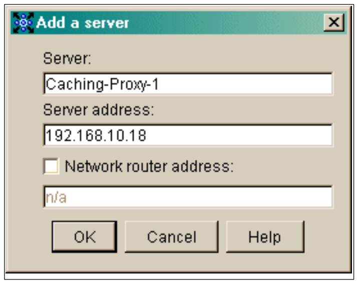


```
Total received since started:
Sent to nonforwarding address:
Processed locally on this machine:
Sent to collocated server:
Forwarded to any cluster:
Not addressed to active cluster/port:
KBytes transferred per second:
Connections per second:
```

```
Heartbeat '192.168.10.14' to '192.168.10.15' successfully added.
ndcontrol highavailability backup add both manual 10009
Backup information successfully added
```

Figure 9-43 Add server

5. Right-click the **Port** item in the left-hand pane and select **Add Server**.



Server: Caching-Proxy-1
Server address: 192.168.10.18
Network router address: n/a

Figure 9-44 Specify server properties

6. The server name can be the real hostname or any name you like. We chose to use more descriptive names (see Figure 9-44). If you do not use the hostname, you must specify the IP address of the server.

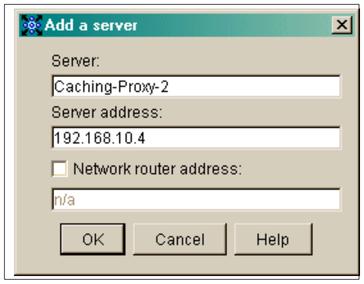
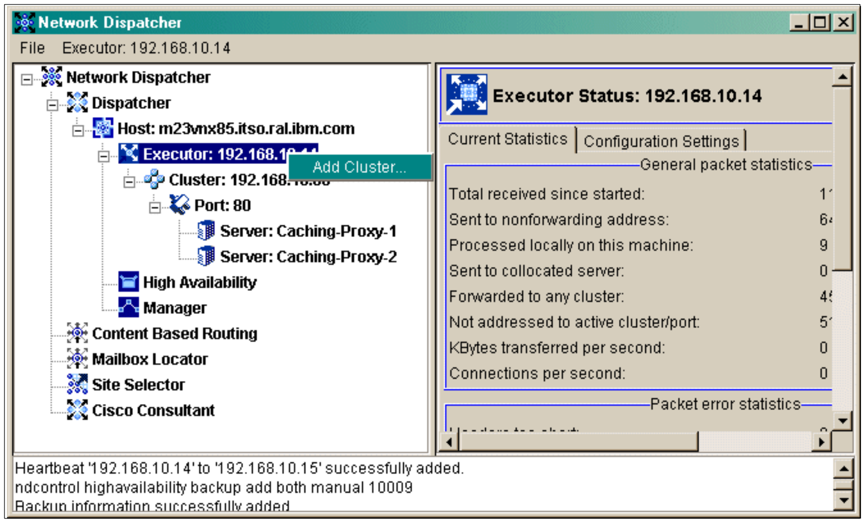


Figure 9-45 Add the second server

7. Repeat steps 5 and 6 for each proxy server.

Create the content server cluster

Now we must create the second load balancer cluster, which will distribute the load between the Web Application servers.

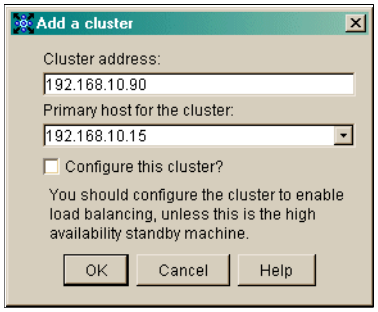


Total received since started: 1
Sent to nonforwarding address: 6
Processed locally on this machine: 9
Sent to collocated server: 0
Forwarded to any cluster: 4
Not addressed to active cluster/port: 5
KBytes transferred per second: 0
Connections per second: 0

```
Heartbeat '192.168.10.14' to '192.168.10.15' successfully added.
ndcontrol highavailability backup add both manual 10009
Backup information successfully added
```

Figure 9-46 Adding the second cluster

1. Right-click the **Executor** item in the left-hand pane and select **Add Cluster**.


Add a cluster

Cluster address:

192.168.10.90

Primary host for the cluster:

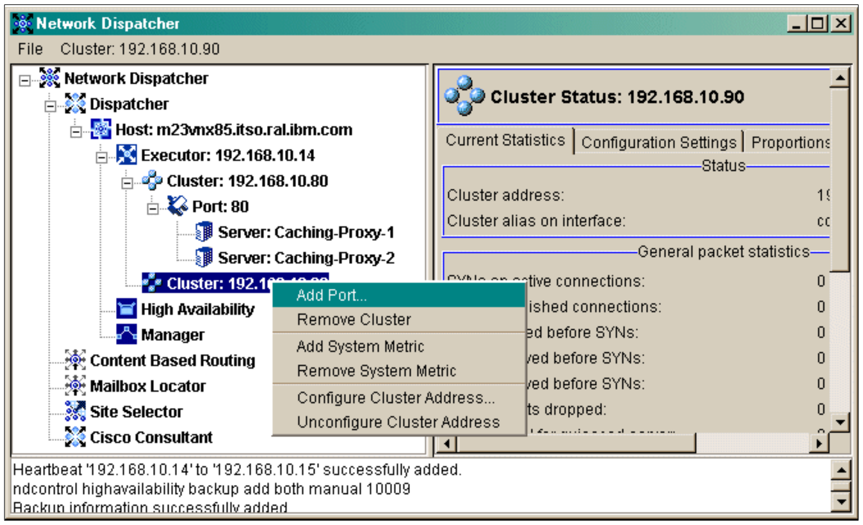
192.168.10.15

Configure this cluster?

You should configure the cluster to enable load balancing, unless this is the high availability standby machine.

Figure 9-47 Cluster properties

2. Specify the cluster address (192.168.10.90) and the primary host (192.168.10.15). Because this is a high availability load balancer configuration, we will use scripts to configure cluster addresses. Therefore, do not select the **Configure this cluster?** box.



```
Heartbeat '192.168.10.14' to '192.168.10.15' successfully added.
ndcontrol highavailability backup add both manual 10009
Backup information successfully added
```

Figure 9-48 Adding a port

3. Right-click the new cluster and select **Add Port**.

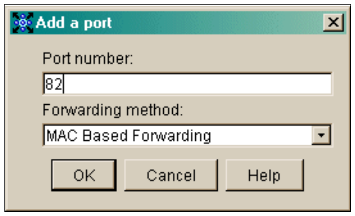
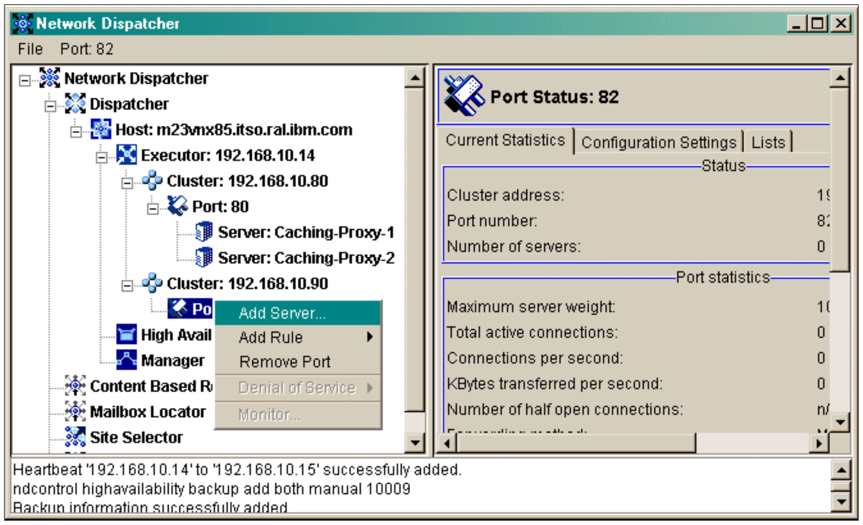


Figure 9-49 Port properties

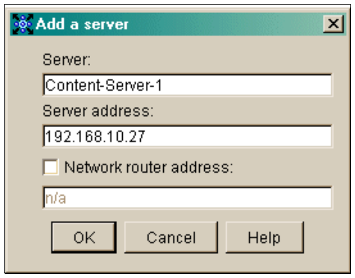
4. This is the port for our content servers. Because we do not want the port to be accessible from the Internet, we specify port 82, which is closed on the first firewall. Note that the port number must match the port number specified in the HTTP server configuration (see "Configure listener" on page 330) and in the caching proxy configuration (see "Basic configuration" on page 331).



```
Heartbeat '192.168.10.14' to '192.168.10.15' successfully added.
ndcontrol highavailability backup add both manual 10009
Backup information successfully added
```

Figure 9-50 Adding a server

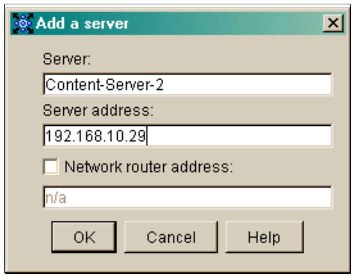
5. Right-click the new port and select **Add Server**.



Server:
Content-Server-1
Server address:
192.168.10.27
☐ Network router address:
n/a

Figure 9-51 Server properties 1

6. Choose a name for the server and enter its IP address.



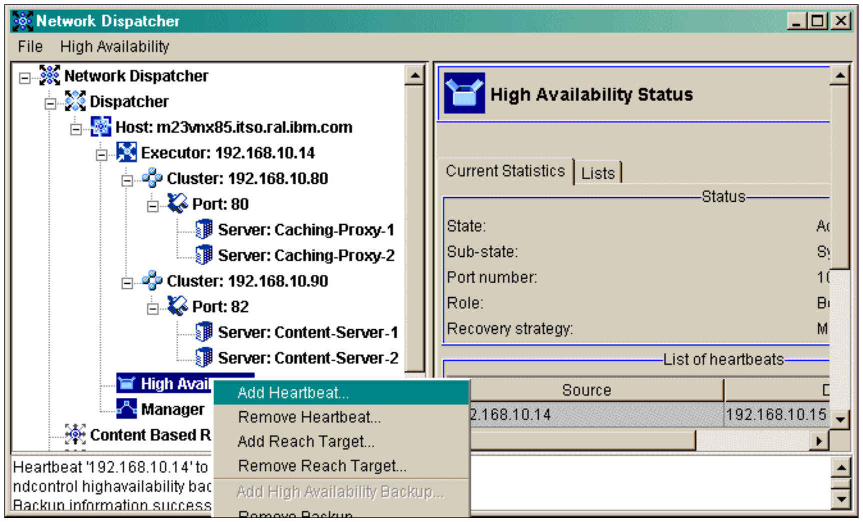
Server:
Content-Server-2
Server address:
192.168.10.29
☐ Network router address:
n/a

Figure 9-52 Server properties 2

7. Repeat steps 6 and 7 for each additional content server.

Configure mutual high availability

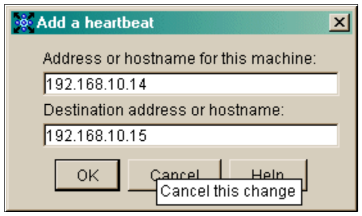
Next, we configure the load balancer nodes for mutual high availability.



Heartbeat '192.168.10.14' to
ndcontrol highavailability bac
Backup information success

Figure 9-53 Adding a heartbeat

1. Right-click **High Availability** in the left-hand pane and select **Add Heartbeat**.

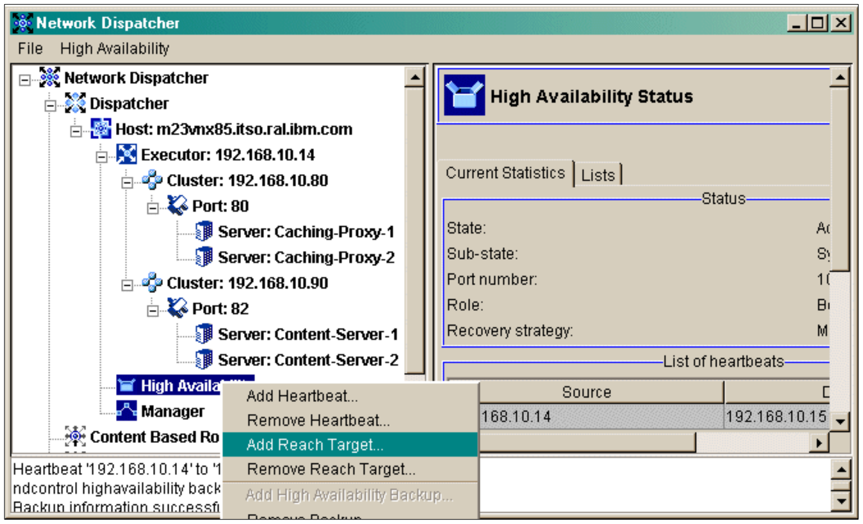


Address or hostname for this machine: 192.168.10.14

Destination address or hostname: 192.168.10.15

Figure 9-54 Heartbeat properties

2. Insert the IP address of the second load balancer node.



Heartbeat '192.168.10.14' to '1
ndcontrol highavailability back
Backup information successfu

Figure 9-55 Adding a reach target

3. Right-click the **High Availability** item again and select **Add Reach Target**.

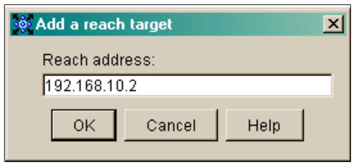
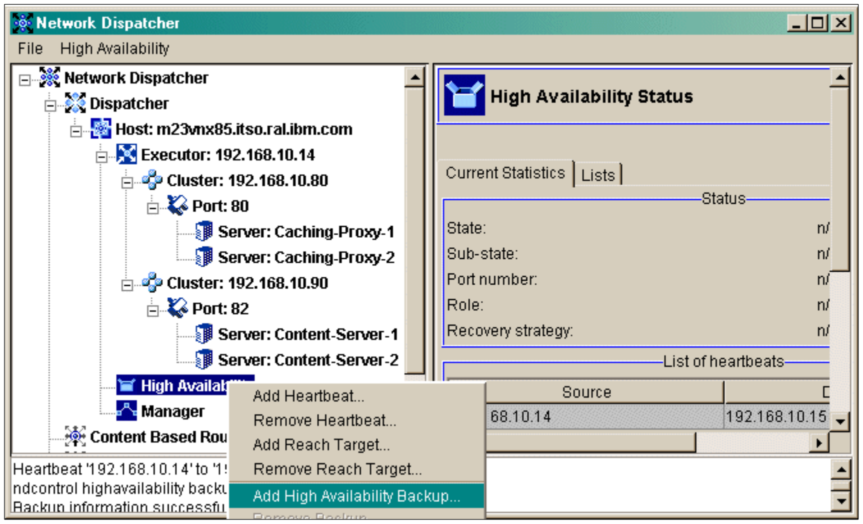


Figure 9-56 Reach target address

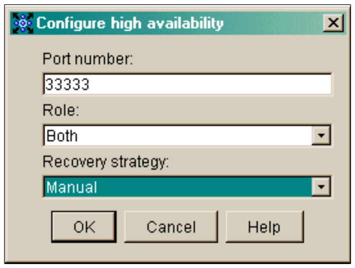
4. The reach target should be a target that is always accessible, a router for example. The reach target is used in conjunction with the heartbeat to ensure full service of the load balancer.



```
Heartbeat '192.168.10.14' to '1
ndcontrol highavailability backu
Backup information successfu
```

Figure 9-57 Adding HA backup

5. Again, right-click the **High Availability** item. This time, select **Add High Availability Backup**.



Port number: 33333

Role: Both

Recovery strategy: Manual

Figure 9-58 HA backup properties

6. Enter a free port number for the communication. Since we are configuring mutual high availability load balancers, select **Both** for the role. If you want to issue manual takeover commands, select **Manual** Recovery strategy.

Edit the takeover scripts

The sample takeover scripts for mutual high availability must be edited to match the chosen cluster IP addresses. Also, you must enter the IP addresses for each of the load balancer nodes. Each load balancer in a mutual high availability configuration must have its own set of takeover scripts. See the following examples for details.

Example 9-2 goActive.cmd for 192.168.10.14

```
@echo off
set CLUSTER1=192.168.10.80
set CLUSTER2=192.168.10.90
set INTERFACE=tr0
set NETMASK=0xffffff00
set PRIM_HOST_1=192.168.10.14
set PRIM_HOST_2=192.168.10.15
rem
if "%1" == "%PRIM_HOST_1%" goto firstclus
if "%1" == "%PRIM_HOST_2%" goto secondclus
goto done

:firstclus
echo "%PRIM_HOST_1% is Active!  Configuring NIC alias(es)..."
call ndconfig lo0 delete %CLUSTER1%
call ndconfig %INTERFACE% alias %CLUSTER1% netmask %NETMASK%
goto done

:secondclus
echo "%PRIM_HOST_1% is in Active state for cluster(s) of %PRIM_HOST_2%"
call ndconfig lo0 delete %CLUSTER2%
call ndconfig %INTERFACE% alias %CLUSTER2% netmask %NETMASK%
:done
```

Example 9-3 goStandby.cmd for 192.168.10.14

```
@echo off
set CLUSTER1=192.168.10.80
set CLUSTER2=192.168.10.90
set INTERFACE=tr0
set NETMASK=0xffffff00
set PRIM_HOST_1=192.168.10.14
set PRIM_HOST_2=192.168.10.15
rem
if "%1" == "%PRIM_HOST_1%" goto firstclus
if "%1" == "%PRIM_HOST_2%" goto secondclus
goto done

:firstclus
   echo "%PRIM_HOST_1% is in Standby state for its cluster(s). "
call ndconfig %INTERFACE% delete %CLUSTER1%
call ndconfig lo0 alias %CLUSTER1% netmask %NETMASK%
goto done
```

```
:secondclus
   echo "%PRIM_HOST_1% is in Standby state for %PRIM_HOST_2% cluster(s)."
call ndconfig %INTERFACE% delete %CLUSTER2%
call ndconfig lo0 alias %CLUSTER2% netmask %NETMASK%
:done
```

Example 9-4 goInOp.cmd for 192.168.10.14

```
@echo off
set PRIM_HOST=192.168.10.14
set CLUSTER=192.168.10.80
set INTERFACE=tr0
rem
echo "Removing loopback and device alias(es)for %PRIM_HOST%"
call ndconfig lo0 delete %CLUSTER%
call ndconfig %INTERFACE% delete %CLUSTER%
```

Example 9-5 goActive.cmd for 192.168.10.15

```
@echo off
set CLUSTER1=192.168.10.90
set CLUSTER2=192.168.10.80
set INTERFACE=tr0
set NETMASK=0xffffff00
set PRIM_HOST_1=192.168.10.15
set PRIM_HOST_2=192.168.10.14
rem
if "%1" == "%PRIM_HOST_1%" goto firstclus
if "%1" == "%PRIM_HOST_2%" goto secondclus
goto done

:firstclus
echo "%PRIM_HOST_1% is Active!  Configuring NIC alias(es)..."
call ndconfig lo0 delete %CLUSTER1%
call ndconfig %INTERFACE% alias %CLUSTER1% netmask %NETMASK%
goto done

:secondclus
echo "%PRIM_HOST_1% is in Active state for cluster(s) of %PRIM_HOST_2%"
call ndconfig lo0 delete %CLUSTER2%
call ndconfig %INTERFACE% alias %CLUSTER2% netmask %NETMASK%
:done
```

Example 9-6 goStandby.cmd for 192.168.10.15

```
@echo off
set CLUSTER1=192.168.10.90
set CLUSTER2=192.168.10.80

set INTERFACE=tr0
set NETMASK=0xffffff00
set PRIM_HOST_1=192.168.10.15
set PRIM_HOST_2=192.168.10.14
rem
if "%1" == "%PRIM_HOST_1%" goto firstclus
if "%1" == "%PRIM_HOST_2%" goto secondclus
goto done

:firstclus
   echo "%PRIM_HOST_1% is in Standby state for its cluster(s). "
call ndconfig %INTERFACE% delete %CLUSTER1%
call ndconfig lo0 alias %CLUSTER1% netmask %NETMASK%
goto done

:secondclus
   echo "%PRIM_HOST_1% is in Standby state for %PRIM_HOST_2% cluster(s)."
call ndconfig %INTERFACE% delete %CLUSTER2%
call ndconfig lo0 alias %CLUSTER2% netmask %NETMASK%
:done
```

Example 9-7 goInOp.cmd for 192.168.10.15

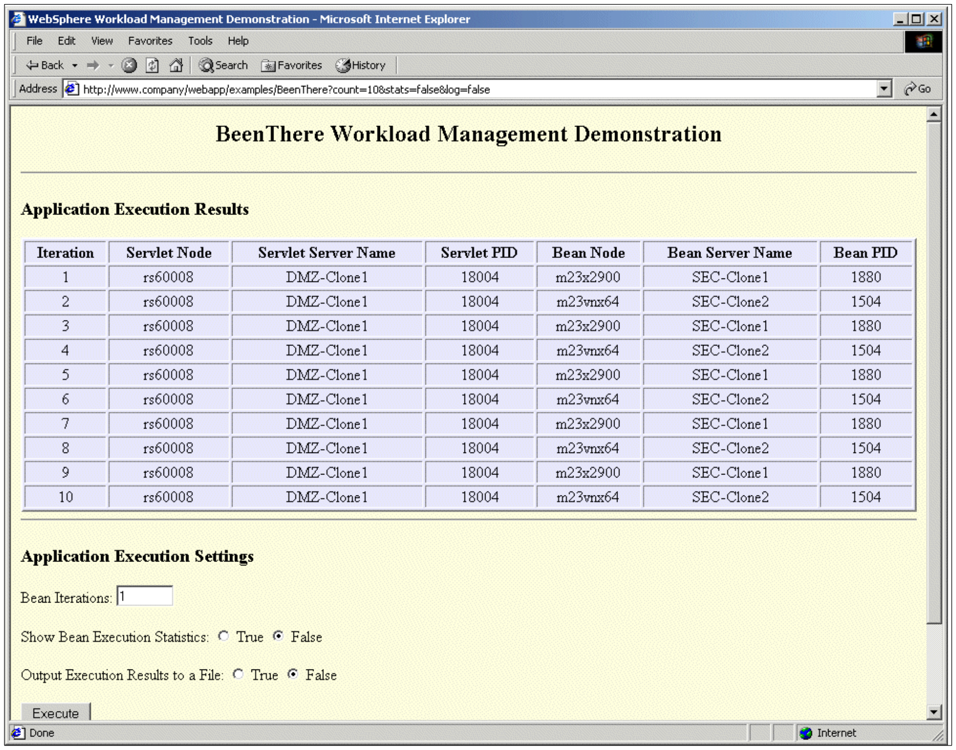
```
@echo off
set PRIM_HOST=192.168.10.15
set CLUSTER=192.168.10.90
set INTERFACE=tr0
rem
echo "Removing loopback and device alias(es)for %PRIM_HOST%"
call ndconfig lo0 delete %CLUSTER%
call ndconfig %INTERFACE% delete %CLUSTER%
```

9.3 Testing the environment

1. Open a browser and go to the URL:

   `http://>yourhostname>/webapp/examples/BeenThere`

2. Enter a number in the iterations field and click the **Execute** button.


WebSphere Workload Management Demonstration - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://www.company/webapp/examples/BeenThere?count=10&stats=false&log=false
Go
Done
Internet

BeenThere Workload Management Demonstration

Application Execution Results

Iteration	Servlet Node	Servlet Server Name	Servlet PID	Bean Node	Bean Server Name	Bean PID
1	rs60008	DMZ-Clone1	18004	m23x2900	SEC-Clone1	1880
2	rs60008	DMZ-Clone1	18004	m23vnx64	SEC-Clone2	1504
3	rs60008	DMZ-Clone1	18004	m23x2900	SEC-Clone1	1880
4	rs60008	DMZ-Clone1	18004	m23vnx64	SEC-Clone2	1504
5	rs60008	DMZ-Clone1	18004	m23x2900	SEC-Clone1	1880
6	rs60008	DMZ-Clone1	18004	m23vnx64	SEC-Clone2	1504
7	rs60008	DMZ-Clone1	18004	m23x2900	SEC-Clone1	1880
8	rs60008	DMZ-Clone1	18004	m23vnx64	SEC-Clone2	1504
9	rs60008	DMZ-Clone1	18004	m23x2900	SEC-Clone1	1880
10	rs60008	DMZ-Clone1	18004	m23vnx64	SEC-Clone2	1504

Application Execution Settings

Bean Iterations: 1

Show Bean Execution Statistics: True [✓ False]

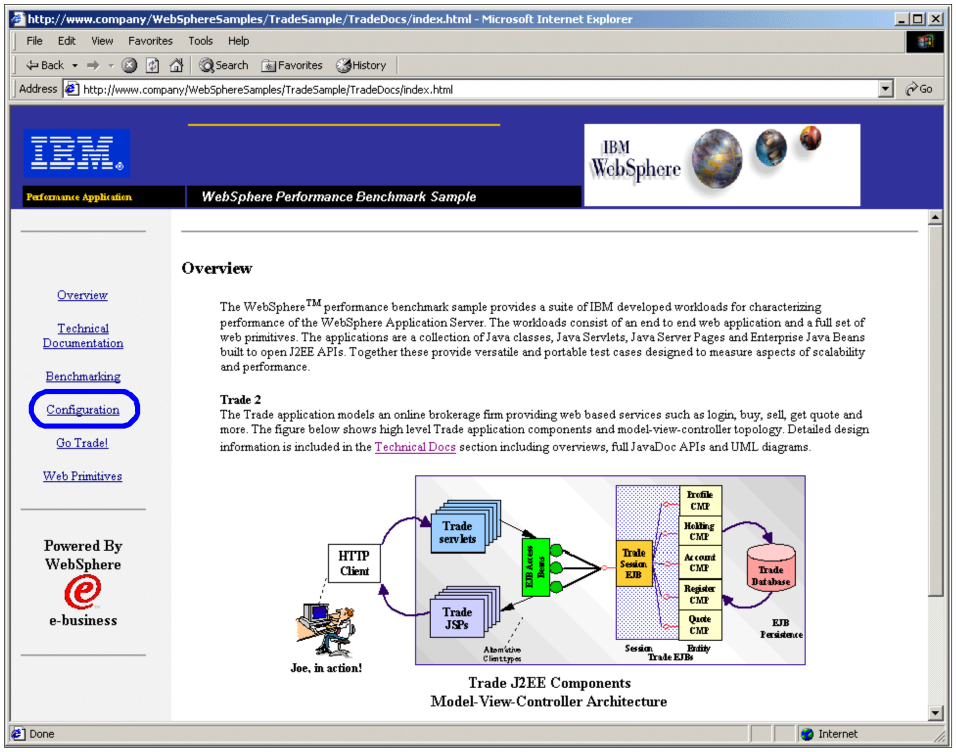
Output Execution Results to a File: True [✓ False]

Execute

Figure 9-59 BeenThere from the sample application

3. In Figure 9-59, you can see that the requests are first processed from a DMZ-Clone and then forwarded to an SEC-Clone. You can also see that load balancing is taking place between the SEC-Clones.
4. Next, go to the Trade Application home page:

   `http://<yourhostname>/WebSphereSamples/TradeSample/TradeDocs/index.html`



Overview

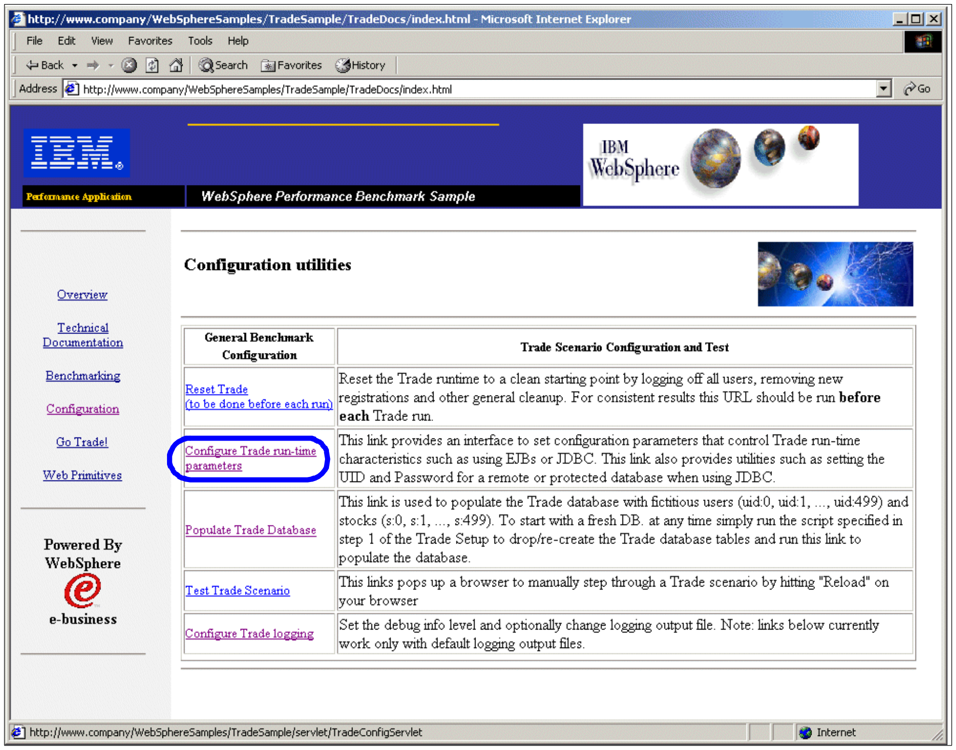
The WebSphere™ performance benchmark sample provides a suite of IBM developed workloads for characterizing performance of the WebSphere Application Server. The workloads consist of an end to end web application and a full set of web primitives. The applications are a collection of Java classes, Java Servlets, Java Server Pages and Enterprise Java Beans built to open J2EE APIs. Together these provide versatile and portable test cases designed to measure aspects of scalability and performance.

Trade 2
The Trade application models an online brokerage firm providing web based services such as login, buy, sell, get quote and more. The figure below shows high level Trade application components and model-view-controller topology. Detailed design information is included in the Technical Docs section including overviews, full JavaDoc APIs and UML diagrams.

Trade J2EE Components
Model-View-Controller Architecture

Figure 9-60 Trade Application home page

5. Click **Configuration**.


http://www.company/WebSphereSamples/TradeSample/TradeDocs/index.html - Microsoft Internet Explorer

Address http://www.company/WebSphereSamples/TradeSample/TradeDocs/index.html

Performance Application

WebSphere Performance Benchmark Sample

Configuration utilities

Overview

Technical Documentation

Benchmarking

Configuration

Go Trade!

Web Primitives

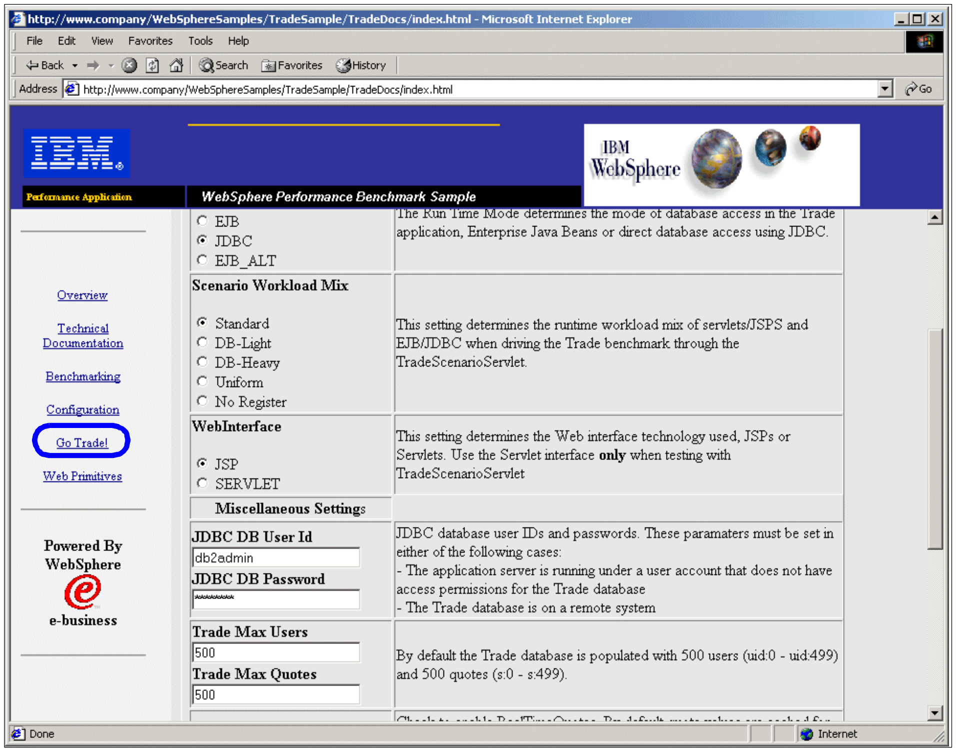
Powered By WebSphere e-business

General Benchmark Configuration	Trade Scenario Configuration and Test
Reset Trade (to be done before each run)	Reset the Trade runtime to a clean starting point by logging off all users, removing new registrations and other general cleanup. For consistent results this URL should be run **before each** Trade run.
Configure Trade run-time parameters	This link provides an interface to set configuration parameters that control Trade run-time characteristics such as using EJBs or JDBC. This link also provides utilities such as setting the UID and Password for a remote or protected database when using JDBC.
Populate Trade Database	This link is used to populate the Trade database with fictitious users (uid:0, uid:1, ..., uid:499) and stocks (s:0, s:1, ..., s:499). To start with a fresh DB. at any time simply run the script specified in step 1 of the Trade Setup to drop/re-create the Trade database tables and run this link to populate the database.
Test Trade Scenario	This links pops up a browser to manually step through a Trade scenario by hitting "Reload" on your browser
Configure Trade logging	Set the debug info level and optionally change logging output file. Note: links below currently work only with default logging output files.

http://www.company/WebSphereSamples/TradeSample/servlet/TradeConfigServlet

Figure 9-61 Trade Application configuration

6. Click **Configure Trade run-time parameters**.


http://www.company/WebSphereSamples/TradeSample/TradeDocs/index.html - Microsoft Internet Explorer

Performance Application | WebSphere Performance Benchmark Sample

Overview
Technical Documentation
Benchmarking
Configuration
Go Trade!
Web Primitives

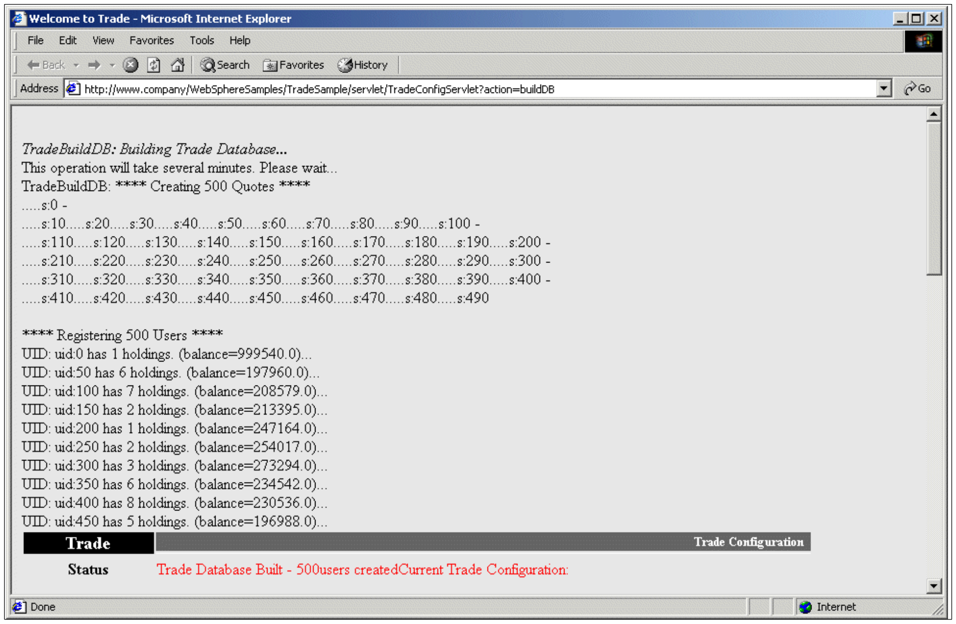
Powered By WebSphere e-business

○ EJB ● JDBC ○ EJB_ALT	The Run Time Mode determines the mode of database access in the Trade application, Enterprise Java Beans or direct database access using JDBC.
Scenario Workload Mix ● Standard ○ DB-Light ○ DB-Heavy ○ Uniform ○ No Register	This setting determines the runtime workload mix of servlets/JSPS and EJB/JDBC when driving the Trade benchmark through the TradeScenarioServlet.
WebInterface ● JSP ○ SERVLET	This setting determines the Web interface technology used, JSPs or Servlets. Use the Servlet interface **only** when testing with TradeScenarioServlet
Miscellaneous Settings	
JDBC DB User Id db2admin **JDBC DB Password** ********	JDBC database user IDs and passwords. These paramaters must be set in either of the following cases: - The application server is running under a user account that does not have access permissions for the Trade database - The Trade database is on a remote system
Trade Max Users 500 **Trade Max Quotes** 500	By default the Trade database is populated with 500 users (uid:0 - uid:499) and 500 quotes (s:0 - s:499).

Done | Internet

Figure 9-62 Trade runtime parameters

7. If your database is remote, specify **JDBC** for the runtime mode.
8. Specify the database user ID and password.
9. Click **Submit**.
10. Now click **Populate Trade Database**. The window shown in Figure 9-63 will be displayed.


Welcome to Trade - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://www.company/WebSphereSamples/TradeSample/servlet/TradeConfigServlet?action=buildDB
Go
Done
Internet

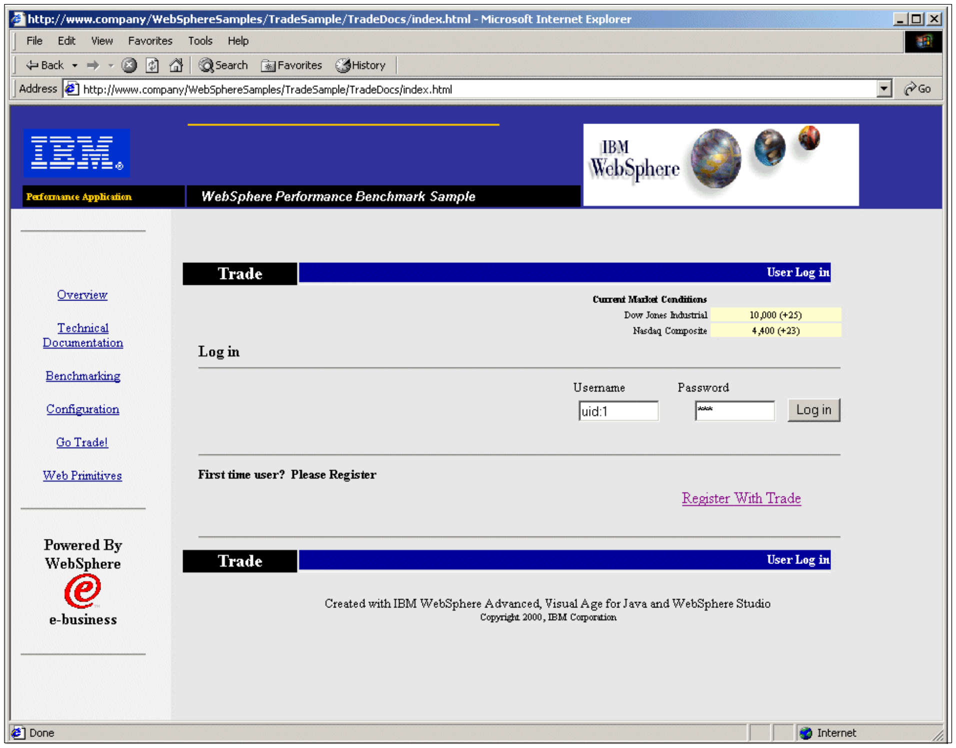
```
TradeBuildDB: Building Trade Database...
This operation will take several minutes. Please wait...
TradeBuildDB: **** Creating 500 Quotes ****
.....s:0 -
.....s:10.....s:20.....s:30.....s:40.....s:50.....s:60.....s:70.....s:80.....s:90.....s:100 -
.....s:110.....s:120.....s:130.....s:140.....s:150.....s:160.....s:170.....s:180.....s:190.....s:200 -
.....s:210.....s:220.....s:230.....s:240.....s:250.....s:260.....s:270.....s:280.....s:290.....s:300 -
.....s:310.....s:320.....s:330.....s:340.....s:350.....s:360.....s:370.....s:380.....s:390.....s:400 -
.....s:410.....s:420.....s:430.....s:440.....s:450.....s:460.....s:470.....s:480.....s:490

**** Registering 500 Users ****
UID: uid:0 has 1 holdings. (balance=999540.0)...
UID: uid:50 has 6 holdings. (balance=197960.0)...
UID: uid:100 has 7 holdings. (balance=208579.0)...
UID: uid:150 has 2 holdings. (balance=213395.0)...
UID: uid:200 has 1 holdings. (balance=247164.0)...
UID: uid:250 has 2 holdings. (balance=254017.0)...
UID: uid:300 has 3 holdings. (balance=273294.0)...
UID: uid:350 has 6 holdings. (balance=234542.0)...
UID: uid:400 has 8 holdings. (balance=230536.0)...
UID: uid:450 has 5 holdings. (balance=196988.0)...
```

Trade **Trade Configuration**

Status Trade Database Built - 500users createdCurrent Trade Configuration:

Figure 9-63 Populating the Trade Database

11. Now you are ready to trade. Click **GoTrade!** as shown in Figure 9-62. The window shown in Figure 9-64 will be displayed.


http://www.company/WebSphereSamples/TradeSample/TradeDocs/index.html - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://www.company/WebSphereSamples/TradeSample/TradeDocs/index.html
Go
IBM
Performance Application
WebSphere Performance Benchmark Sample
IBM WebSphere
Overview
Technical Documentation
Benchmarking
Configuration
Go Trade!
Web Primitives
Powered By WebSphere
e-business
Trade
User Log in
Log in
Username
Password
Log in
Register With Trade
Done
Internet

Current Market Conditions

Dow Jones Industrial	10,000 (+25)
Nasdaq Composite	4,400 (+23)

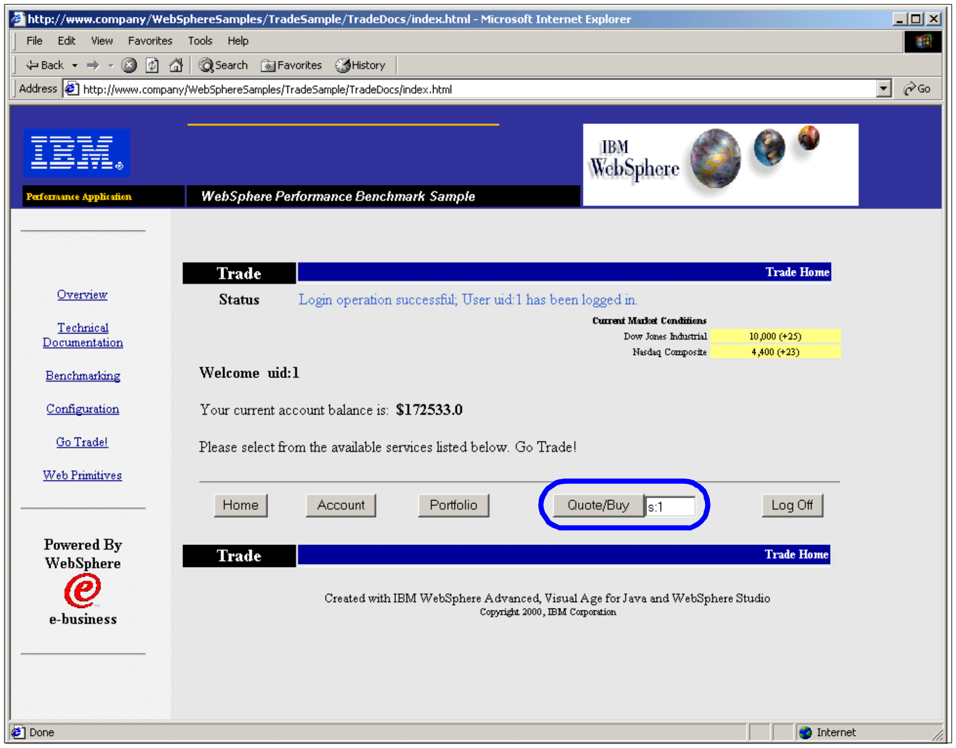
Username: uid:1
Password: ***

First time user? Please Register

Created with IBM WebSphere Advanced, Visual Age for Java and WebSphere Studio
Copyright 2000, IBM Corporation

Figure 9-64 Trade application logon

12. Accept the default password and click **Logon**.


http://www.company/WebSphereSamples/TradeSample/TradeDocs/index.html - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://www.company/WebSphereSamples/TradeSample/TradeDocs/index.html
Go
IBM
Performance Application
WebSphere Performance Benchmark Sample
IBM WebSphere
Overview
Technical Documentation
Benchmarking
Configuration
Go Trade!
Web Primitives
Powered By WebSphere
e-business
Trade
Trade Home
Home
Account
Portfolio
Quote/Buy
s:1
Log Off
Done
Internet

Status Login operation successful; User uid:1 has been logged in.

Current Market Conditions

Dow Jones Industrial	10,000 (+25)
Nasdaq Composite	4,400 (+23)

Welcome uid:1

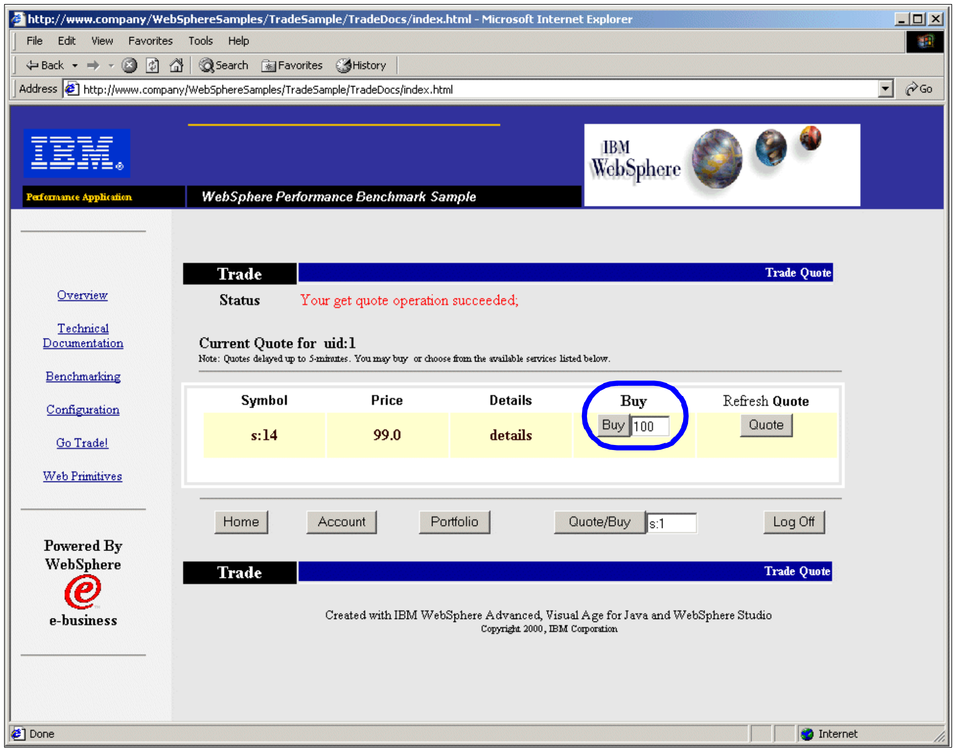
Your current account balance is: **$172533.0**

Please select from the available services listed below. Go Trade!

Created with IBM WebSphere Advanced, Visual Age for Java and WebSphere Studio
Copyright 2000, IBM Corporation

Figure 9-65 Trade application status window

13. Enter a number in the Quote/Buy field and click **Quote/Buy**.


Trade | Trade Quote

Status Your get quote operation succeeded;

Current Quote for uid:1

Note: Quotes delayed up to 5-minutes. You may buy or choose from the available services listed below.

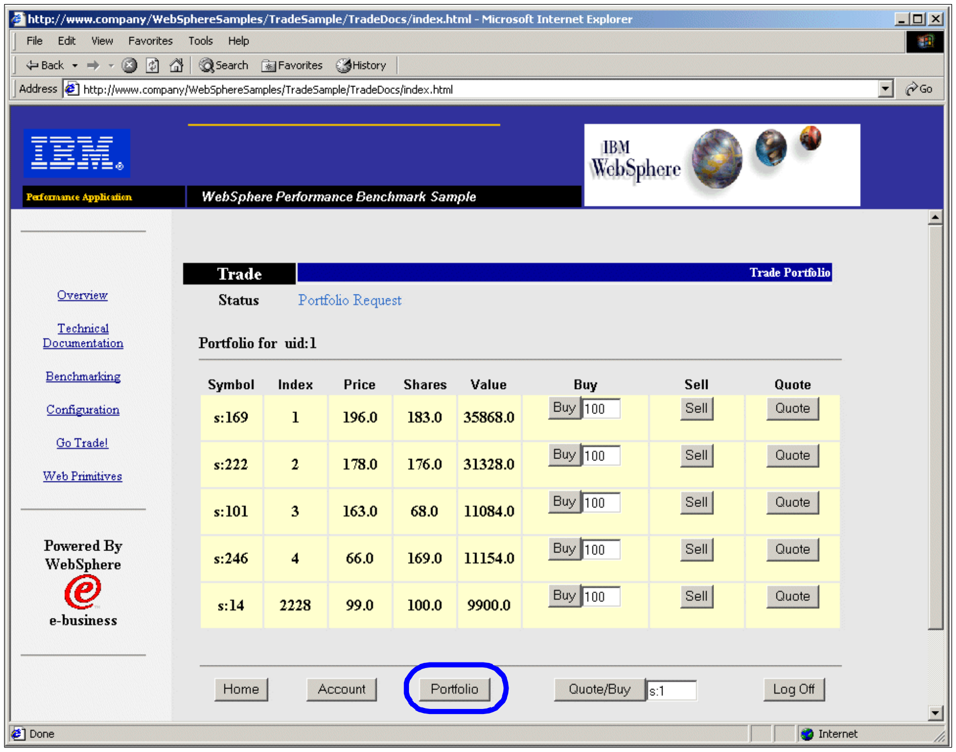
Symbol	Price	Details	Buy	Refresh Quote
s:14	99.0	details	Buy 100	Quote

Created with IBM WebSphere Advanced, Visual Age for Java and WebSphere Studio
Copyright 2000, IBM Corporation

http://www.company/WebSphereSamples/TradeSample/TradeDocs/index.html - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://www.company/WebSphereSamples/TradeSample/TradeDocs/index.html
Go
Performance Application
WebSphere Performance Benchmark Sample
IBM WebSphere
Overview
Technical Documentation
Benchmarking
Configuration
Go Trade!
Web Primitives
Powered By WebSphere
e-business
Home
Account
Portfolio
Quote/Buy
s:1
Log Off
Done
Internet

Figure 9-66 Trade application stock quote

14.Click **Buy** to purchase the offered stock.


http://www.company/WebSphereSamples/TradeSample/TradeDocs/index.html - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://www.company/WebSphereSamples/TradeSample/TradeDocs/index.html
Go
IBM
Performance Application
WebSphere Performance Benchmark Sample
IBM WebSphere
Overview
Technical Documentation
Benchmarking
Configuration
Go Trade!
Web Primitives
Powered By WebSphere
e-business
Trade
Trade Portfolio
Home
Account
Portfolio
Quote/Buy
s:1
Log Off
Done
Internet

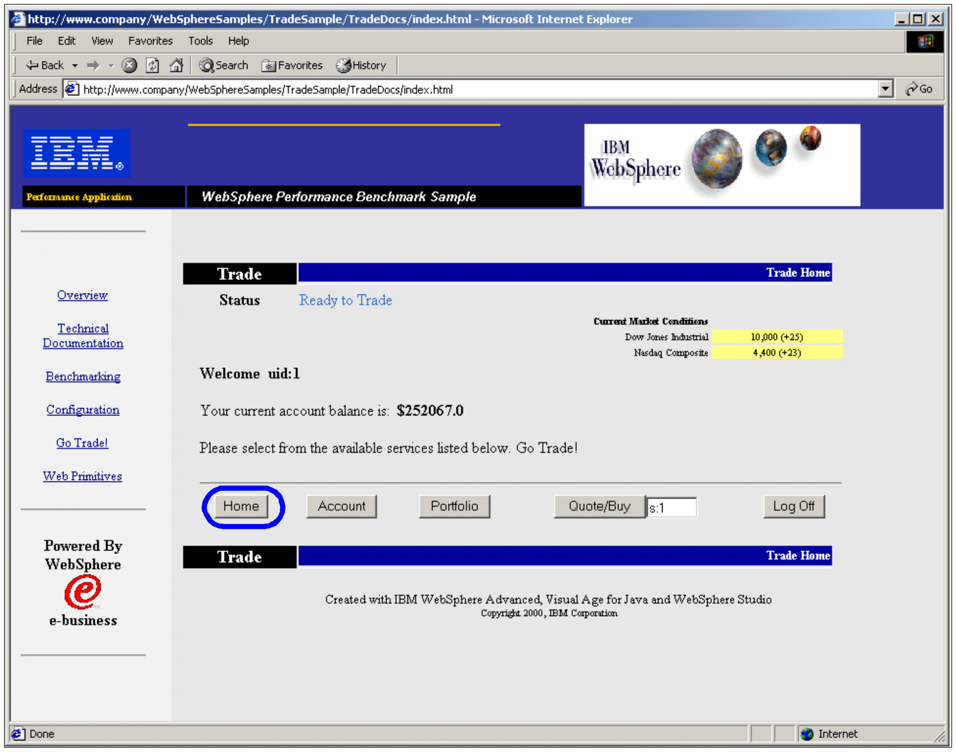
Status Portfolio Request

Portfolio for uid:1

Symbol	Index	Price	Shares	Value	Buy	Sell	Quote
s:169	1	196.0	183.0	35868.0	Buy 100	Sell	Quote
s:222	2	178.0	176.0	31328.0	Buy 100	Sell	Quote
s:101	3	163.0	68.0	11084.0	Buy 100	Sell	Quote
s:246	4	66.0	169.0	11154.0	Buy 100	Sell	Quote
s:14	2228	99.0	100.0	9900.0	Buy 100	Sell	Quote

Figure 9-67 Trade application portfolio

15. Get a few quotes, buy and sell some stock.
16. Click **Portfolio** to get a listing of your current stocks.


http://www.company/WebSphereSamples/TradeSample/TradeDocs/index.html - Microsoft Internet Explorer

WebSphere Performance Benchmark Sample

Trade Status Ready to Trade

Current Market Conditions

Dow Jones Industrial	10,000 (+25)
Nasdaq Composite	4,400 (+23)

Welcome uid:1

Your current account balance is: $252067.0

Please select from the available services listed below. Go Trade!

Created with IBM WebSphere Advanced, Visual Age for Java and WebSphere Studio
Copyright 2000, IBM Corporation

File Edit View Favorites Tools Help
Back Search Favorites History
Address http://www.company/WebSphereSamples/TradeSample/TradeDocs/index.html Go
IBM
Performance Application
IBM WebSphere
Trade Home
Overview
Technical Documentation
Benchmarking
Configuration
Go Trade!
Web Primitives
Home Account Portfolio Quote/Buy s:1 Log Off
Powered By WebSphere
e-business
Done
Internet

Figure 9-68 Trade application user home page

17.Click **Home** to get your account balance.

A

Configuration files

This appendix contains configuration files for the example environment described in Chapter 8., “Sample ASP environment” on page 227.

ASP environment configuration files

Primary load balancer node

Example 9-8 ASP-Environment.cfg

```
ndcontrol set loglevel 1
ndcontrol executor start

ndcontrol cluster add 192.168.10.60 primaryhost 192.168.10.14
ndcontrol cluster set 192.168.10.60 proportions 45 45 10 0
ndcontrol port add 192.168.10.60:80
ndcontrol server add 192.168.10.60:80:Caching-Proxy-2 address 192.168.10.4
ndcontrol server add 192.168.10.60:80:Caching-Proxy-1 address 192.168.10.18

ndcontrol cluster add 192.168.10.70 primaryhost 192.168.10.14
ndcontrol cluster set 192.168.10.70 proportions 45 45 10 0
ndcontrol port add 192.168.10.70:80
ndcontrol server add 192.168.10.70:80:HTTP-Server-1 address 192.168.10.27
ndcontrol server add 192.168.10.70:80:HTTP-Server-2 address 192.168.10.29

ndcontrol manager start manager.log 10004
ndcontrol advisor start Http 192.168.10.70:80 TradeCluster.log
ndcontrol advisor start Ping 192.168.10.60:80 TradeProxy.log

ndcontrol highavailability heartbeat add 192.168.10.14 192.168.10.15
ndcontrol highavailability backup add primary=192.168.10.14 manual 10009
ndcontrol highavailability reach add 192.168.10.12
```

Backup load balancer node

Example 9-9 ASP-Environment.cfg

```
ndcontrol set loglevel 1
ndcontrol executor start

ndcontrol cluster add 192.168.10.60 primaryhost 192.168.10.15
ndcontrol cluster set 192.168.10.60 proportions 45 45 10 0
ndcontrol port add 192.168.10.60:80
ndcontrol server add 192.168.10.60:80:Caching-Proxy-2 address 192.168.10.4
ndcontrol server add 192.168.10.60:80:Caching-Proxy-1 address 192.168.10.18

ndcontrol cluster add 192.168.10.70 primaryhost 192.168.10.15
ndcontrol cluster set 192.168.10.70 proportions 45 45 10 0
ndcontrol port add 192.168.10.70:80
ndcontrol server add 192.168.10.70:80:HTTP-Server-1 address 192.168.10.27
ndcontrol server add 192.168.10.70:80:HTTP-Server-2 address 192.168.10.29
ndcontrol manager start manager.log 10004
ndcontrol advisor start Http 192.168.10.70:80 TradeCluster.log
ndcontrol advisor start Ping 192.168.10.60:80 TradeProxy.log

ndcontrol highavailability heartbeat add 192.168.10.15 192.168.10.14
ndcontrol highavailability backup add backup=192.168.10.15 manual 10009
ndcontrol highavailability reach add 192.168.10.12
```

Web Application server node

Example 9-10   ASP-Example-cfg.xml

```
<?xml version="1.0"?>
<!DOCTYPE websphere-sa-config SYSTEM
"file:///$XMLConfigDTDLocation$$dsep$xmlconfig.dtd" >
<websphere-sa-config>
    <virtual-host action="update" name="foo24host">
...
skipping the mime table...
...
        <alias-list>
            <alias>foo24host:81</alias>
            <alias>foo24host.itso.ral.ibm.com:81</alias>
        </alias-list>
    </virtual-host>
    <virtual-host action="update" name="fakehost">
...
skipping the mime table...
...
<alias-list>
            <alias>fakehost:81</alias>
            <alias>fakehost.itso.ral.ibm.com:81</alias>
        </alias-list>
    </virtual-host>
    <jms-provider action="update" name="IBM MQSeries">
        <description>After MQSeries is installed on one or more nodes
            this provider may be used by specifying the classpath on the
            nodes tab of the property notebook in the Admin Console or
            with the install command of wscp or xmlconfig.</description>

<external-initial-context-factory>com.ibm.websphere.naming.WsnInitialContextFac
tory</external-initial-context-factory>
        <external-provider-url>iiop://localhost</external-provider-url>
        <jndi-binding-mechanism></jndi-binding-mechanism>
    </jms-provider>
    <url-provider action="update" name="Default URL Provider">
        <description>This is the URL Provider for the protocols
            supported by the JDK (e.g., http, file, ftp, etc.).  The
            protocol and stream handler class name properties are not
            used for this provider.</description>
        <protocol>unused</protocol>
        <stream-handler-class-name>unused</stream-handler-class-name>
        <install-info>
            <node-name>rs60008</node-name>
            <jar-file-location>unused</jar-file-location>
        </install-info>
        <install-info>
            <node-name>rs600035</node-name>
            <jar-file-location>unused</jar-file-location>
        </install-info>
    </url-provider>
    <jdbc-driver action="update" name="Sample DB Driver">

<implementation-class>COM.ibm.db2.jdbc.DB2ConnectionPoolDataSource</implementat
ion-class>
        <description>Example Data Source</description>
        <data-source action="update" name="foo24_DS">
            <database-name>foo24db</database-name>
            <minimum-pool-size>1</minimum-pool-size>
            <maximum-pool-size>10</maximum-pool-size>
            <connection-timeout>180</connection-timeout>
            <idle-timeout>1800</idle-timeout>
            <orphan-timeout>1800</orphan-timeout>
            <statement-cache-size>100</statement-cache-size>
            <default-user>db2admin</default-user>
            <default-password>{xor}Oz1tPjsyNjE=</default-password>

<disable-auto-connectioncleanup>false</disable-auto-connectioncleanup>
            <description>DataSource for Customer www.foo24trade</description>
            <jndi-name>jdbc/foo24_DS</jndi-name>
            <config-properties/>
        </data-source>
        <data-source action="update" name="fake_DS">
            <database-name>fakeDB</database-name>
            <minimum-pool-size>1</minimum-pool-size>
            <maximum-pool-size>10</maximum-pool-size>
            <connection-timeout>180</connection-timeout>
            <idle-timeout>1800</idle-timeout>
            <orphan-timeout>1800</orphan-timeout>
            <statement-cache-size>100</statement-cache-size>
            <default-user>db2admin</default-user>
            <default-password>{xor}Oz1tPjsyNjE=</default-password>

<disable-auto-connectioncleanup>false</disable-auto-connectioncleanup>
            <description>DataSource for Customer www.faketrade</description>
            <jndi-name>jdbc/fake_DS</jndi-name>
            <config-properties/>
        </data-source>
        <install-info>
            <node-name>rs60008</node-name>

<jdbc-zipfile-location>/home/db2inst1//sqllib/java12/db2java.zip</jdbc-zipfile-
location>
        </install-info>
        <install-info>
            <node-name>rs600035</node-name>

<jdbc-zipfile-location>/home/db2inst1//sqllib/java12/db2java.zip</jdbc-zipfile-
location>
        </install-info>
    </jdbc-driver>
    <node action="update" name="rs600035"/>
    <node action="update" name="rs60008"/>
    <security-config security-cache-timeout="600" security-enabled="false">
        <app-security-defaults>
            <realm-name>Default</realm-name>
            <challenge-type ssl-enabled="false">
                <basic-challenge/>
            </challenge-type>
        </app-security-defaults>
        <auth-mechanism>
            <localos>
                <user-id></user-id>
                <password>{xor}</password>
            </localos>
        </auth-mechanism>
        <ssl-config>

<key-file-name>${WAS_HOME}/etc/DummyServerKeyFile.jks</key-file-name>
            <key-file-password>{xor}CDo9Hgw=</key-file-password>
            <key-file-format>0</key-file-format>
            <client-authentication>false</client-authentication>
            <security-level>0</security-level>
            <crypto-hardware-enabled>false</crypto-hardware-enabled>
            <crypto-library-file></crypto-library-file>
            <crypto-password>{xor}</crypto-password>
            <crypto-token-type></crypto-token-type>

<trust-file-name>${WAS_HOME}/etc/DummyServerTrustFile.jks</trust-file-name>
            <trust-file-password>{xor}CDo9Hgw=</trust-file-password>
            <property name="com.ibm.ssl.protocol" value="SSLv3"/>
        </ssl-config>
    </security-config>
    <server-group action="update" name="foo24ServerGroup">
        <server-group-attributes>
            <executable>java</executable>
            <command-line-arguments/>
            <environment/>
            <user-id></user-id>
            <group-id></group-id>
            <working-directory></working-directory>
            <umask>18</umask>
            <stdin></stdin>
            <stdout>stdout.txt</stdout>
            <stderr>stderr.txt</stderr>
            <process-priority>20</process-priority>
            <maximum-startup-attempts>2</maximum-startup-attempts>
            <ping-interval>60</ping-interval>
            <ping-timeout>200</ping-timeout>
            <ping-initial-timeout>600</ping-initial-timeout>
            <selection-policy>roundrobinpreferlocal</selection-policy>
            <trace-specification></trace-specification>
            <trace-output></trace-output>
            <transaction-log-file></transaction-log-file>
            <olt-enabled>false</olt-enabled>
            <system-properties/>
            <debug-enabled>false</debug-enabled>
            <transaction-timeout>120</transaction-timeout>

<transaction-inactivity-timeout>60000</transaction-inactivity-timeout>
            <thread-pool-size>20</thread-pool-size>
            <module-visibility>3</module-visibility>

<use-domain-qualified-user-names>false</use-domain-qualified-user-names>
            <default-data-source></default-data-source>
            <security-enabled>false</security-enabled>
            <cache-config>
                <cache-size>2047</cache-size>
                <cache-sweep-interval>1000</cache-sweep-interval>
                <passivation-directory></passivation-directory>
            </cache-config>
            <log-file-spec></log-file-spec>
            <olt-server-host>localhost</olt-server-host>
            <olt-server-port>2102</olt-server-port>
            <selection-policy>roundrobinpreferlocal</selection-policy>
            <source-path></source-path>
            <thread-pool-size>20</thread-pool-size>
            <jvm-config>
                <initial-heap-size>0</initial-heap-size>
                <max-heap-size>0</max-heap-size>
                <system-properties/>

<additional-command-line-arguments></additional-command-line-arguments>
                <debug-mode>false</debug-mode>
                <debug-string></debug-string>
                <run-hprof>false</run-hprof>
                <hprof-args></hprof-args>
                <disable-jit>false</disable-jit>
                <verbose-class-loading>false</verbose-class-loading>
                <verbose-jni>false</verbose-jni>
                <verbose-gc>false</verbose-gc>
                <boot-classpath-replace></boot-classpath-replace>
                <boot-classpath-append></boot-classpath-append>
                <boot-classpath-prepend></boot-classpath-prepend>
            </jvm-config>
            <web-container>
                <dynamic-cache-config>
                    <enabled>false</enabled>
                    <cache-size>1000</cache-size>
                    <default-priority>1</default-priority>
                </dynamic-cache-config>
                <transport name="http">
                    <transport-host>*</transport-host>
                    <transport-port>9080</transport-port>
                    <http-transport>
                        <connection-timeout>5</connection-timeout>
                        <backlog-connections>511</backlog-connections>
                        <keep-alive-timeout>5</keep-alive-timeout>
                        <maximum-keep-alive>25</maximum-keep-alive>
                        <maximum-req-keep-alive>100</maximum-req-keep-alive>
                        <ssl-enabled>false</ssl-enabled>
                    </http-transport>
                </transport>
                <thread-maximum-size>50</thread-maximum-size>
                <thread-minimum-size>10</thread-minimum-size>
                <thread-inactivity-timeout>10</thread-inactivity-timeout>
                <thread-is-growable>false</thread-is-growable>
                <session-manager>

<enable-security-integration>false</enable-security-integration>
                    <enable-ssl-tracking>false</enable-ssl-tracking>

<invalidation-schedule-first-hour>0</invalidation-schedule-first-hour>

<invalidation-schedule-second-hour>0</invalidation-schedule-second-hour>
                    <persistent-db2-row-size>0</persistent-db2-row-size>

<maximum-inmemory-session-count>1000</maximum-inmemory-session-count>
                    <write-contents>0</write-contents>
                    <write-frequency>0</write-frequency>
                    <write-interval>120</write-interval>
                    <enable-sessions>false</enable-sessions>
                    <enable-url-rewriting>false</enable-url-rewriting>
                    <enable-cookies>true</enable-cookies>
```

```
<enable-protocol-switch-rewriting>false</enable-protocol-switch-rewriting>
                    <cookie name="JSESSIONID">
                        <domain></domain>
                        <maximum>-1</maximum>
                        <path>/</path>
                        <secure>false</secure>
                    </cookie>
                    <tuning-invalidation-time>30</tuning-invalidation-time>
                    <persistent-sessions>false</persistent-sessions>
                    <data-source name="">
                        <default-user></default-user>
                        <default-password>{xor}</default-password>
                    </data-source>
                    <persistent-table-space-name/>
                    <using-multi-row>false</using-multi-row>
                    <allow-overflow>true</allow-overflow>
                </session-manager>
            </web-container>
            <orb-config>
                <bootstrap-host-name/>
                <bootstrap-port>900</bootstrap-port>
                <comm-trace-enabled>false</comm-trace-enabled>
                <connection-cache-maximum>240</connection-cache-maximum>
                <connection-cache-minimum>100</connection-cache-minimum>
                <external-config-url/>
                <force-tunnel>whenrequired</force-tunnel>
                <listener-port>0</listener-port>
                <locate-request-timeout>180</locate-request-timeout>
                <local-host-name/>
                <lsd-host-name/>
                <no-local-copies>false</no-local-copies>
                <request-retries-count>1</request-retries-count>
                <request-retries-delay>0</request-retries-delay>
                <request-timeout>180</request-timeout>
                <thread-pool-size>20</thread-pool-size>
                <tunnel-agent-url/>
                <rmi-remote-code-base/>
                <ssl-listener-port>0</ssl-listener-port>
            </orb-config>
            <custom-service-config-list/>
        </server-group-attributes>
        <clone name="foo24Clone-1">
            <parent-node>rs60008</parent-node>
        </clone>
        <clone name="foo24clone-2">
            <parent-node>rs600035</parent-node>
        </clone>
    </server-group>
    <server-group action="update" name="fakeServerGroup">
        <server-group-attributes>
            <executable>java</executable>
            <command-line-arguments/>
            <environment/>
            <user-id></user-id>
            <group-id></group-id>
            <working-directory></working-directory>
            <umask>18</umask>
            <stdin></stdin>
            <stdout>stdout.txt</stdout>
            <stderr>stderr.txt</stderr>
            <process-priority>20</process-priority>
            <maximum-startup-attempts>2</maximum-startup-attempts>
            <ping-interval>60</ping-interval>
            <ping-timeout>200</ping-timeout>
            <ping-initial-timeout>600</ping-initial-timeout>
            <selection-policy>roundrobinpreferlocal</selection-policy>
            <trace-specification></trace-specification>
            <trace-output></trace-output>
            <transaction-log-file></transaction-log-file>
            <olt-enabled>false</olt-enabled>
            <system-properties/>
            <debug-enabled>false</debug-enabled>
            <transaction-timeout>120</transaction-timeout>

<transaction-inactivity-timeout>60000</transaction-inactivity-timeout>
            <thread-pool-size>20</thread-pool-size>
            <module-visibility>3</module-visibility>

<use-domain-qualified-user-names>false</use-domain-qualified-user-names>
            <default-data-source></default-data-source>
            <security-enabled>false</security-enabled>
            <cache-config>
                <cache-size>2047</cache-size>
                <cache-sweep-interval>1000</cache-sweep-interval>
                <passivation-directory></passivation-directory>
            </cache-config>
            <log-file-spec></log-file-spec>
            <olt-server-host>localhost</olt-server-host>
            <olt-server-port>2102</olt-server-port>
            <selection-policy>roundrobinpreferlocal</selection-policy>
            <source-path></source-path>
            <thread-pool-size>20</thread-pool-size>
            <jvm-config>
                <initial-heap-size>0</initial-heap-size>
                <max-heap-size>0</max-heap-size>
                <system-properties/>
```

```
<additional-command-line-arguments></additional-command-line-arguments>
                <debug-mode>false</debug-mode>
                <debug-string></debug-string>
                <run-hprof>false</run-hprof>
                <hprof-args></hprof-args>
                <disable-jit>false</disable-jit>
                <verbose-class-loading>false</verbose-class-loading>
                <verbose-jni>false</verbose-jni>
                <verbose-gc>false</verbose-gc>
                <boot-classpath-replace></boot-classpath-replace>
                <boot-classpath-append></boot-classpath-append>
                <boot-classpath-prepend></boot-classpath-prepend>
            </jvm-config>
            <web-container>
                <dynamic-cache-config>
                    <enabled>false</enabled>
                    <cache-size>1000</cache-size>
                    <default-priority>1</default-priority>
                </dynamic-cache-config>
                <transport name="http">
                    <transport-host>*</transport-host>
                    <transport-port>9080</transport-port>
                    <http-transport>
                        <connection-timeout>5</connection-timeout>
                        <backlog-connections>511</backlog-connections>
                        <keep-alive-timeout>5</keep-alive-timeout>
                        <maximum-keep-alive>25</maximum-keep-alive>
                        <maximum-req-keep-alive>100</maximum-req-keep-alive>
                        <ssl-enabled>false</ssl-enabled>
                    </http-transport>
                </transport>
                <thread-maximum-size>50</thread-maximum-size>
                <thread-minimum-size>10</thread-minimum-size>
                <thread-inactivity-timeout>10</thread-inactivity-timeout>
                <thread-is-growable>false</thread-is-growable>
                <session-manager>

<enable-security-integration>false</enable-security-integration>
                    <enable-ssl-tracking>false</enable-ssl-tracking>

<invalidation-schedule-first-hour>0</invalidation-schedule-first-hour>

<invalidation-schedule-second-hour>0</invalidation-schedule-second-hour>
                    <persistent-db2-row-size>0</persistent-db2-row-size>

<maximum-inmemory-session-count>1000</maximum-inmemory-session-count>
                    <write-contents>0</write-contents>
                    <write-frequency>0</write-frequency>
                    <write-interval>120</write-interval>
                    <enable-sessions>false</enable-sessions>
                    <enable-url-rewriting>false</enable-url-rewriting>
                    <enable-cookies>true</enable-cookies>

<enable-protocol-switch-rewriting>false</enable-protocol-switch-rewriting>
                    <cookie name="JSESSIONID">
                        <domain></domain>
                        <maximum>-1</maximum>
                        <path>/</path>
                        <secure>false</secure>
                    </cookie>
                    <tuning-invalidation-time>30</tuning-invalidation-time>
                    <persistent-sessions>false</persistent-sessions>
                    <data-source name="">
                        <default-user></default-user>
                        <default-password>{xor}</default-password>
                    </data-source>
                    <persistent-table-space-name/>
                    <using-multi-row>false</using-multi-row>
                    <allow-overflow>true</allow-overflow>
                </session-manager>
            </web-container>
            <orb-config>
                <bootstrap-host-name/>
                <bootstrap-port>900</bootstrap-port>
                <comm-trace-enabled>false</comm-trace-enabled>
                <connection-cache-maximum>240</connection-cache-maximum>
                <connection-cache-minimum>100</connection-cache-minimum>
                <external-config-url/>
                <force-tunnel>whenrequired</force-tunnel>
                <listener-port>0</listener-port>
                <locate-request-timeout>180</locate-request-timeout>
                <local-host-name/>
                <lsd-host-name/>
                <no-local-copies>false</no-local-copies>
                <request-retries-count>1</request-retries-count>
                <request-retries-delay>0</request-retries-delay>
                <request-timeout>180</request-timeout>
                <thread-pool-size>20</thread-pool-size>
                <tunnel-agent-url/>
                <rmi-remote-code-base/>
                <ssl-listener-port>0</ssl-listener-port>
            </orb-config>
            <custom-service-config-list/>
        </server-group-attributes>
        <clone name="fakeClone-1">
            <parent-node>rs60008</parent-node>
        </clone>
        <clone name="fakeClone-2">
            <parent-node>rs600035</parent-node>
        </clone>
    </server-group>
    <enterprise-application action="create" name="foo24TradeApp">
        <source-node>rs600035</source-node>

<ear-file-name>/usr/websphere/appserver/installableApps/TradeSample.ear</ear-fi
le-name>
        <enterprise-app-install-info>
            <node-name>rs60008</node-name>

<ear-install-directory>/usr/websphere/appserver/installedApps/foo24TradeApp.ear
</ear-install-directory>
        </enterprise-app-install-info>
        <enterprise-app-install-info>
            <node-name>rs600035</node-name>

<ear-install-directory>/usr/websphere/appserver/installedApps/foo24TradeApp.ear
</ear-install-directory>
        </enterprise-app-install-info>
        <application-binding>
            <authorization-table/>
            <run-as-map/>
        </application-binding>
        <ejb-module name="Trade Sample EJB">
            <jar-file>TradeEJBs.jar</jar-file>
            <module-install-info>
                <server-group-name>foo24ServerGroup</server-group-name>
            </module-install-info>
            <ejb-module-binding>
                <data-source>
                    <jndi-name>jdbc/foo24_DS</jndi-name>
                    <default-user>db2admin</default-user>
                    <default-password>{xor}Oz1tPjsyNjE=</default-password>
                </data-source>
                <enterprise-bean-binding name="Sesion_1_Bnd">
                    <jndi-name>trade/TradeHome</jndi-name>
                    <ejb-ref-binding name="EjbRefBinding_1">
                        <jndi-name>trade/AccountHome</jndi-name>
                    </ejb-ref-binding>
                    <ejb-ref-binding name="EjbRefBinding_2">
                        <jndi-name>trade/HoldingHome</jndi-name>
                    </ejb-ref-binding>
                    <ejb-ref-binding name="EjbRefBinding_3">
                        <jndi-name>trade/KeySequenceHome</jndi-name>
                    </ejb-ref-binding>
                    <ejb-ref-binding name="EjbRefBinding_4">
                        <jndi-name>trade/KeysEntityHome</jndi-name>
                    </ejb-ref-binding>
                    <ejb-ref-binding name="EjbRefBinding_5">
                        <jndi-name>trade/ProfileHome</jndi-name>
                    </ejb-ref-binding>
                    <ejb-ref-binding name="EjbRefBinding_6">
                        <jndi-name>trade/QuoteHome</jndi-name>
                    </ejb-ref-binding>
                    <ejb-ref-binding name="EjbRefBinding_7">
                        <jndi-name>trade/TradeRegistryHome</jndi-name>
                    </ejb-ref-binding>
                    <ejb-ref-binding name="EjbRefBinding_8">
                        <jndi-name>trade/TradeHome</jndi-name>
                    </ejb-ref-binding>
                </enterprise-bean-binding>
                <enterprise-bean-binding name="Sesion_2_Bnd">
                    <jndi-name>trade/KeySequenceHome</jndi-name>
                    <ejb-ref-binding name="EjbRefBinding_9">
                        <jndi-name>trade/KeysEntityHome</jndi-name>
                    </ejb-ref-binding>
                </enterprise-bean-binding>
                <enterprise-bean-binding name="ContainerManagedEntity_1_Bnd">
                    <jndi-name>trade/ProfileHome</jndi-name>
                </enterprise-bean-binding>
                <enterprise-bean-binding name="ContainerManagedEntity_2_Bnd">
                    <jndi-name>trade/HoldingHome</jndi-name>
                </enterprise-bean-binding>
                <enterprise-bean-binding name="ContainerManagedEntity_5_Bnd">
                    <jndi-name>trade/QuoteHome</jndi-name>
                </enterprise-bean-binding>
                <enterprise-bean-binding name="ContainerManagedEntity_3_Bnd">
                    <jndi-name>trade/AccountHome</jndi-name>
                </enterprise-bean-binding>
                <enterprise-bean-binding name="ContainerManagedEntity_4_Bnd">
                    <jndi-name>trade/TradeRegistryHome</jndi-name>
                </enterprise-bean-binding>
                <enterprise-bean-binding name="ContainerManagedEntity_6_Bnd">
                    <jndi-name>trade/KeysEntityHome</jndi-name>
                </enterprise-bean-binding>
            </ejb-module-binding>
        </ejb-module>
        <web-module name="theme Web Application">
            <war-file>theme.war</war-file>
            <context-root>/tradetheme</context-root>
            <module-install-info>
                <server-group-name>foo24ServerGroup</server-group-name>
            </module-install-info>
            <web-module-binding>
                <virtual-host-name>foo24host</virtual-host-name>
            </web-module-binding>
```

```
            </web-module>
            <web-module name="trade Web Application">
                <war-file>Trade.war</war-file>
                <context-root>/WebSphereSamples/TradeSample</context-root>
                <module-install-info>
                    <server-group-name>foo24ServerGroup</server-group-name>
                </module-install-info>
                <web-module-binding>
                    <virtual-host-name>foo24host</virtual-host-name>
                    <ejb-ref-binding name="EjbRefBinding_1">
                        <jndi-name>trade/AccountHome</jndi-name>
                    </ejb-ref-binding>
                    <ejb-ref-binding name="EjbRefBinding_2">
                        <jndi-name>trade/HoldingHome</jndi-name>
                    </ejb-ref-binding>
                    <ejb-ref-binding name="EjbRefBinding_3">
                        <jndi-name>trade/KeySequenceHome</jndi-name>
                    </ejb-ref-binding>
                    <ejb-ref-binding name="EjbRefBinding_4">
                        <jndi-name>trade/KeysEntityHome</jndi-name>
                    </ejb-ref-binding>
                    <ejb-ref-binding name="EjbRefBinding_5">
                        <jndi-name>trade/ProfileHome</jndi-name>
                    </ejb-ref-binding>
                    <ejb-ref-binding name="EjbRefBinding_6">
                        <jndi-name>trade/QuoteHome</jndi-name>
                    </ejb-ref-binding>
                    <ejb-ref-binding name="EjbRefBinding_7">
                        <jndi-name>trade/TradeRegistryHome</jndi-name>
                    </ejb-ref-binding>
                    <ejb-ref-binding name="EjbRefBinding_8">
                        <jndi-name>trade/TradeHome</jndi-name>
                    </ejb-ref-binding>
                    <resource-ref-binding name="ResourceRefBinding_1">
                        <jndi-name>jdbc/foo24_DS</jndi-name>
                        <basic-auth-data>
                            <user-id/>
                            <password/>
                        </basic-auth-data>
                    </resource-ref-binding>
                </web-module-binding>
            </web-module>
        </enterprise-application>
        <enterprise-application action="create" name="fakeTradeApp">
            <source-node>rs600035</source-node>

    <ear-file-name>/usr/websphere/appserver/installableApps/TradeSample.ear</ear-fi
    le-name>
            <enterprise-app-install-info>
                <node-name>rs60008</node-name>

    <ear-install-directory>/usr/websphere/appserver/installedApps/fakeTradeApp.ear<
    /ear-install-directory>
            </enterprise-app-install-info>
            <enterprise-app-install-info>
                <node-name>rs600035</node-name>

    <ear-install-directory>/usr/websphere/appserver/installedApps/fakeTradeApp.ear<
    /ear-install-directory>
            </enterprise-app-install-info>
            <application-binding>
                <authorization-table/>
                <run-as-map/>
            </application-binding>
            <ejb-module name="Trade Sample EJB">
                <jar-file>TradeEJBs.jar</jar-file>
                <module-install-info>
                    <server-group-name>fakeServerGroup</server-group-name>
                </module-install-info>
                <ejb-module-binding>
                    <data-source>
                        <jndi-name>jdbc/fake_DS</jndi-name>
                        <default-user>db2admin</default-user>
                        <default-password>{xor}Oz1tPjsyNjE=</default-password>
                    </data-source>
                    <enterprise-bean-binding name="Sesion_1_Bnd">
                        <jndi-name>trade/TradeHome</jndi-name>
                        <ejb-ref-binding name="EjbRefBinding_1">
                            <jndi-name>trade/AccountHome</jndi-name>
                        </ejb-ref-binding>
                        <ejb-ref-binding name="EjbRefBinding_2">
                            <jndi-name>trade/HoldingHome</jndi-name>
                        </ejb-ref-binding>
                        <ejb-ref-binding name="EjbRefBinding_3">
                            <jndi-name>trade/KeySequenceHome</jndi-name>
                        </ejb-ref-binding>
                        <ejb-ref-binding name="EjbRefBinding_4">
                            <jndi-name>trade/KeysEntityHome</jndi-name>
                        </ejb-ref-binding>
                        <ejb-ref-binding name="EjbRefBinding_5">
                            <jndi-name>trade/ProfileHome</jndi-name>
                        </ejb-ref-binding>
                        <ejb-ref-binding name="EjbRefBinding_6">
                            <jndi-name>trade/QuoteHome</jndi-name>
                        </ejb-ref-binding>
                        <ejb-ref-binding name="EjbRefBinding_7">
                            <jndi-name>trade/TradeRegistryHome</jndi-name>
                        </ejb-ref-binding>
                        <ejb-ref-binding name="EjbRefBinding_8">
                            <jndi-name>trade/TradeHome</jndi-name>
                        </ejb-ref-binding>
                    </enterprise-bean-binding>
                    <enterprise-bean-binding name="Sesion_2_Bnd">
                        <jndi-name>trade/KeySequenceHome</jndi-name>
                        <ejb-ref-binding name="EjbRefBinding_9">
                            <jndi-name>trade/KeysEntityHome</jndi-name>
                        </ejb-ref-binding>
                    </enterprise-bean-binding>
                    <enterprise-bean-binding name="ContainerManagedEntity_1_Bnd">
                        <jndi-name>trade/ProfileHome</jndi-name>
                    </enterprise-bean-binding>
                    <enterprise-bean-binding name="ContainerManagedEntity_2_Bnd">
                        <jndi-name>trade/HoldingHome</jndi-name>
                    </enterprise-bean-binding>
                    <enterprise-bean-binding name="ContainerManagedEntity_5_Bnd">
                        <jndi-name>trade/QuoteHome</jndi-name>
                    </enterprise-bean-binding>
                    <enterprise-bean-binding name="ContainerManagedEntity_3_Bnd">
                        <jndi-name>trade/AccountHome</jndi-name>
                    </enterprise-bean-binding>
                    <enterprise-bean-binding name="ContainerManagedEntity_4_Bnd">
                        <jndi-name>trade/TradeRegistryHome</jndi-name>
                    </enterprise-bean-binding>
                    <enterprise-bean-binding name="ContainerManagedEntity_6_Bnd">
                        <jndi-name>trade/KeysEntityHome</jndi-name>
                    </enterprise-bean-binding>
                </ejb-module-binding>
            </ejb-module>
            <web-module name="theme Web Application">
                <war-file>theme.war</war-file>
                <context-root>/tradetheme</context-root>
                <module-install-info>
                    <server-group-name>fakeServerGroup</server-group-name>
                </module-install-info>
                <web-module-binding>
                    <virtual-host-name>fakehost</virtual-host-name>
                </web-module-binding>
            </web-module>
            <web-module name="trade Web Application">
                <war-file>Trade.war</war-file>
                <context-root>/WebSphereSamples/TradeSample</context-root>
                <module-install-info>
                    <server-group-name>fakeServerGroup</server-group-name>
                </module-install-info>
                <web-module-binding>
                    <virtual-host-name>fakehost</virtual-host-name>
                    <ejb-ref-binding name="EjbRefBinding_1">
```

```
                    <jndi-name>trade/AccountHome</jndi-name>
                </ejb-ref-binding>
                <ejb-ref-binding name="EjbRefBinding_2">
                    <jndi-name>trade/HoldingHome</jndi-name>
                </ejb-ref-binding>
                <ejb-ref-binding name="EjbRefBinding_3">
                    <jndi-name>trade/KeySequenceHome</jndi-name>
                </ejb-ref-binding>
                <ejb-ref-binding name="EjbRefBinding_4">
                    <jndi-name>trade/KeysEntityHome</jndi-name>
                </ejb-ref-binding>
                <ejb-ref-binding name="EjbRefBinding_5">
                    <jndi-name>trade/ProfileHome</jndi-name>
                </ejb-ref-binding>
                <ejb-ref-binding name="EjbRefBinding_6">
                    <jndi-name>trade/QuoteHome</jndi-name>
                </ejb-ref-binding>
                <ejb-ref-binding name="EjbRefBinding_7">
                    <jndi-name>trade/TradeRegistryHome</jndi-name>
                </ejb-ref-binding>
                <ejb-ref-binding name="EjbRefBinding_8">
                    <jndi-name>trade/TradeHome</jndi-name>
                </ejb-ref-binding>
                <resource-ref-binding name="ResourceRefBinding_1">
                    <jndi-name>jdbc/fake_DS</jndi-name>
                    <basic-auth-data>
                        <user-id/>
                        <password/>
                    </basic-auth-data>
                </resource-ref-binding>
            </web-module-binding>
        </web-module>
    </enterprise-application>
</websphere-sa-config>
```

Glossary

3270 Web bridge. A feature of CICS Web Support which Web enables CICS transactions written for 3270 terminal input/output without modification of the source code or screen maps. It is an ideal solution for an existing application that needs to be accessed from the Web, especially if the code is too fragile to be changed or is not available.

A

ACK. The ACK control bit, which occupies no sequence space, acknowledges receipt of all previous communication by specifying the next sequence number that is expected.

address. A unique code assigned to each device or workstation connected to a network. A standard IP address is a 32– bit address field containing two parts: the network address, and the host number.

Administration Console. The Administration Console (graphical interface) is a Java application that is used to perform configuration, administration and tracing tasks, as well as integrating and configuring new profiles and transcoders.

advisor. Advisors are a function of the Dispatcher component that collect and analyze feedback from individual servers and inform the manager function about server load.

affinity. The process of identifying the types of processing that a server can perform and directing a processing request to that server based on the processing type.

agent. (1) In systems management, a user that, for a particular interaction, has assumed an agent role. (2) An entity that represents one or more managed objects by (a) emitting notifications regarding the objects and (b) handling requests from managers for management operations to modify or query the objects.

alias. An additional name assigned to a server. The alias makes the server independent of the name of its host machine. The alias must be defined in the domain name server.

AOR (application-owning region). A CICS region in a MRO environment that “owns” the CICS applications, and invokes them on behalf of remotely attached terminal (or Web) users. See also *TOR*, FOR and *listener region.*

API (Application Programming Interface). Application programming interface. A set of calling conventions defining how a service is invoked through a software package.

APPC (Advanced Program-to-Program communication). An implementation of the SNA LU 6.2 protocol that allows interconnected systems to communicate and share the processing of programs.

Applet. A Java applet is a small application program that is downloaded to and executed on a Web browser or network computer. A Java applet typically performs the type of operations that client code would perform in a client/server architecture. It edits input, controls the screen, and communicates transactions to a server, which in turn performs the data or database operations.

B

backup. When the Dispatcher load balancing component is configured for high availability, a backup is the partner of the primary machine. The backup machine monitors the status of the primary machine and takes over if necessary.

bandwidth. The difference between the highest and lowest frequencies of a transmission channel; the amount of data that can be sent through a given communication circuit per second.

begin range. In rules-based load balancing, the lower value specified in a rule. The default for this value depends on the type of rule**.**

BLI (business logic interface). An externally callable interface provided by CICS Web support. It allows a client to invoke the business logic in a CICS application. It is implemented by the module DFHWBBLI. It provides a mechanism for implementing Web-aware presentation logic in the converter. The converter provides Decode and Encode routines to receive and send the HTTP presentation logic.

BMP (bean-managed persistence) bean. An enterprise bean that manages its own persistence.

browser. An application that displays World Wide Web documents, usually referred to as a Web browser.

C

Cache Server. Some networks use a cache server to store Web pages and other data, so that if the same pages are requested frequently, they can be served from the cache rather than repeatedly retrieved from external Web servers. The external cache is an HTTP proxy such as IBM Web Traffic Express. IBM WebSphere Transcoding Publisher can use it to store and retrieve transcoded Web pages and intermediate results to avoid repeating the transcoding of frequently accessed pages and giving thus better performance.

cache. A local store of content that is managed by a proxy or an external cache manager.

Caching Proxy. A caching proxy server that can help speed up end-user response time through efficient caching schemes. The WebSphere Edge Server Caching Proxy also can be used as a filtering point and to enable other Edge Server functions.

CBR. Content Based Routing. A component of Network Dispatcher, CBR works with the Caching Proxy to load balance incoming requests based on the content of the requests.

cbrcontrol. Provides the interface to the Content Based Router component of Network Dispatcher.

cbrserver. In CBR, handles the requests from the executor, manager and advisors.

CCF (Common Connector Framework). IBM's common client programming model for connectors. These interfaces allow VisualAge for Java's Enterprise Access Builder for transactions to easily build Java applets or servlets to access programs or transactions in a CICS region. The CCF also provides a common infrastructure programming model for connectors, which gives a component environment such as WebSphere a standard view of a connector.

CGI (Common Gateway Interface). A defined standard that describes how a Web Server can communicate with another application (the CGI program) on the same machine. Usually a CGI program is a small program that takes data from a Web server and does something with it, like execute a database query.

CICS (Customer Information Control System). A distributed on-line transaction processing system designed to support a network of many terminals. The CICS family of products is available for a variety of platforms ranging from a single workstation to the largest mainframe.

CICS CORBA client support. A feature which allows a client on any platform, including a Web browser or a Web server, to access a CICS Java application, provided the request conforms to CORBA (Common Object Request Broker Architecture) standards. CICS Transaction Server V1.3 also provides support for writing new CICS applications in Java and for wrapping existing applications with this interface.

CICS Gateway for Java. The CICS Gateway for Java has been superseded by the CICS Transaction Gateway.

CICS Universal Client. A program which provides an API for connectivity to CICS from non-System/390 server platforms, including OS/2, Windows NT, Windows 95 and 98, AIX and Solaris. It can be used from a server program written in most of the languages that these platforms support, including C, C++, COBOL, PL/I, and REXX. It is also available to programs using object classes for C++, the Microsoft COM interface, and VisualAge, Visual C++, Visual Basic, and PowerBuilder.

CIG (CICS Internet Gateway). An early method of Web-enabling 3270 transactions provided with the CICS Client. The CICS Internet Gateway has now been superseded by the CICS Transaction Gateway Terminal Servlet.

client. A computer system or process that requests a service of another computer system or process. For example, a workstation or personal computer requesting HTML documents from a Web server is a client of that server.

As in client-server computing, the application that makes requests to the server and, often, handles the interaction necessary with the user.

client-server computing. A form of distributed processing, in which the task required to be processed is accomplished by a client portion that requests services and a server portion that fulfills those requests. The client and server remain transparent to each other in terms of location and platform. See *client* and *server*.

cluster address. In the Dispatcher, the address to which clients connect.

cluster. In the Dispatcher load balancing component, a cluster is a group of TCP or UDP servers that are used for the same purpose and are identified by a single host name.

clustered server. A server that the Dispatcher groups with other servers into a single virtual server. Network Dispatcher balances TCP or UDP traffic among these clustered servers.

Clustering. Clustering is a technique used to provide scalability through the use of multiple copies of an application on the same or separate machines. Careful management of the different applications is necessary to ensure that they work together effectively. WebSphere has limited clustering support in Version 2.x and more support in Version 3.0.

COMMAREA. A buffer or data area used by a CICS program to hold input and output data when LINKed (called) as a subroutine from another CICS program.

commit. An action that an transaction processing monitor takes, to make permanent the changes it has made to recoverable resources during a logical unit of work.

Connectors. The term connectors, or e-business connectors, is used to describe gateway products from IBM that allow access to enterprise data on back-end systems over the Internet. They include direct browser access to back-end systems such as DB2 through Net.data and also Java access through products such as the CICS Gateway for Java. See `http://www.ibm.com/software/ebusiness/connectors.html` for more information.

CORBA (Common Object Request Broker Architecture). A cross-platform, industry-standard distributed object protocol. CORBA is used to locate and use objects on a variety of platforms, written in a variety of languages across a network. See `http://www.omg.org` for more information on CORBA.

cross port affinity. Cross port affinity is the affinity (sticky) feature expanded to cover across multiple ports.

CTG (CICS Transaction Gateway). A set of software components which enable CICS transaction and programs to be invoked from Java programs.

CWI (CICS Web Interface). A set of resources supplied with CICS/ESA V4.1 and CICS TS V1.2 that allows CICS programs to be invoked from the Internet. Superseded by *CICS Web support.*

CWS (CICS Web support). A set of resources supplied with CICS TS V1.3 that provide CICS with functionality similar to a real Web server, and allows both CICS programs and 3270 transactions to be invoked from the Internet. See also *CICS Web Interface*.

D

Dispatcher. A component of WebSphere Edge Server Load Balancer (Network Dispatcher) that balances TCP or UDP traffic among groups of individual linked servers in a cluster. The Dispatcher machine is the server running the Dispatcher code. Also referred to as the Dispatcher component of Load Balancer.

E

EAB (Enterprise Access Builder). A framework and tools provided by VisualAge for Java, that work in conjunction with connectors such as the CICS Transaction Gateway (CICS connector) to allow you to access the function and data within Enterprise Information Systems (such as CICS).

e-business. e-business is a term used by IBM to describe the use of Internet technologies to transform business processes. What this means in practice is using Internet clients such as Web browsers as front ends for applications that access back-end legacy systems to allow greater access. See `http://www.software.ibm.com/ebusiness` for more information.

ECI (External Call Interface). An application programming interface (API) provided by the CICS client that enables an application to call a CICS program as a subroutine. The client application communicates with the server CICS program using a data area called a COMMAREA.

enterprise bean. An EJB component that implements a business task or business entity.

Enterprise Java Beans. Despite the name, Enterprise Java Beans or EJBs are not Java Beans. Enterprise Java Beans are server-side Java components that are designed for distributed environments. They do not exist in isolation but are rather deployed in containers that provide services such as security, naming and directory services and persistent storage. WebSphere Application Server is just such a container. See `http://java.sun.com/products/ejb/` for more information.

entity bean. An enterprise bean that represents persistent data maintained in a database. An entity bean can manage its own persistence or it can delegate this function to its container. An entity bean is identified by its primary key. If the container in which an entity bean is hosted crashes, the entity bean, its primary key, and its remote references survive the crash

EPI (External Presentation Interface). An application programming interface (API) provided by the CICS Client that allows a 3270 based CICS transaction to be driven programmatically.

ESI (External Security Interface). An application programming interface (API) provided by the CICS Universal Client that utilizes the APPC PEM support in a CICS region to verify and update userids and passwords.

ESM (External Security Manager). A external security product, such as RACF, used by CICS for authentication and authorization of users.

EXCI (External CICS Interface). An application programming interface (API) provided by CICS on OS/390 that enables a non-CICS application to call a CICS program as a subroutine. It is similar in function to the ECI but is available on OS/390. The EXCI is available in all supported releases of CICS and is extended in CICS Transaction Server Version 1.3 to allow multiple calls to CICS programs within a single CICS unit of work.

executor. One of several Dispatcher functions. The executor routes requests to the TCP or UDP servers, and also monitors the number of new, active, and finished connections and does garbage collection of completed or reset connections. The executor supplies the new and active connections to the manager function. In Cisco Consultant, the executor holds configuration information and contains the information required to connect to the Cisco CSS Switch.

F

FIN state. The status of a transaction that has finished. Once a transaction is in FIN state, the Network Dispatcher garbage collector can clear the memory reserved for the connection.

FIN. A control bit occupying one sequence number, which indicates that the sender will send no more data or control that occupies sequence space. The code FIN is taken from the word finis.

firewall. A computer that connects a private network, such as a business, to a public network, such as the Internet. It contains programs that limit access between two networks.

forward proxy. A network arrangement in which a Caching Proxy server is used by client browsers as a gateway to the Internet.

G

GRE. Generic Routing Encapsulation. A protocol which allows an arbitrary network protocol A to be transmitted over any other arbitrary protocol B, by encapsulating the packets of A within GRE packets, which in turn are contained within packets of B.

H

HACL (Host Access Class Library). This is provided by IBM's Host On-Demand software as a set of Java classes and methods that allow the development of platform-independent applications that can access host information (such as CICS 3270 terminals) at the data stream level.

heartbeat. A simple packet sent between two Dispatcher machines in high availability mode used by the standby Dispatcher to monitor the health of the active Dispatcher.

high availability. A Dispatcher feature in which one Dispatcher can take over the function of another, if that part fails.

host name. The symbolic name assigned to a host. Host names are resolved to IP addresses through a domain name server.

Host On-Demand. IBM's terminal emulation software, downloadable "on-demand" from a Web server. Part of the IBM WebSphere Host Integration suite of products.

Host Publisher. An IBM program product which allows the integration of multiple sources of data including host and database applications as a single Web page, with no change to backend systems. Part of the IBM WebSphere Host Integration suite of products.

host. A computer, connected to a network, that provides an access point to that network. A host can be a client, a server, or both simultaneously.

HTML. Hypertext Markup Language. The language used to create hypertext documents. Hypertext documents include links to other documents that contain additional information about the highlighted term or subject. HTML controls the format of text and position of form input areas, for example, as well as the navigable links.

HTTP. Hypertext Transfer Protocol. The protocol used to transfer and display hypertext documents.

I

ICMP. Internet Control Message Protocol. A message control and error-reporting protocol between a host server and a gateway to the Internet.

IIOP. Internet Inter ORB Protocol (IIOP) is an Internet protocol used for CORBA object communication. For more information see `http://www.whatis.com/iiop.htm`.

Image Transcoder. Image Transcoder is the transcoder that can scale, modify quality, and modify color levels in JPEG and GIF images. Additionally, the Image Transcoder can convert JPEGs to GIFs for devices that do not render JPEGs.

IMAP. Internet Message Access Protocol. A protocol allowing a client to access and manipulate electronic mail messages on a server. It permits manipulation of remote message folders (mailboxes) in a way that is functionally equivalent to local mailboxes.

Intermediary. In a typical Web environment, information is simply sent from a server to a browser for display and interaction. However, there are many ways that adding an intermediary between the browser and the server can improve the system. For example, an intermediary can keep track of the information the user has viewed in order to make it easier to find information again. Or, an intermediary may enhance the information that the user sees by adding annotations and personalization beyond what the server was designed to do. Intermediaries turn the network into a "smart pipe" with applications that can enhance the information on the Web.

Internet. The worldwide collection of interconnected networks that use the Internet suite of protocols and permit public access.

intranet. A secure, private network that integrates Internet standards and applications (such as Web browsers) with an organization's existing computer networking infrastructure.

IP address. Internet Protocol address. The unique 32-bit address that specifies the actual location of each device or workstation in a network. It is also known as an Internet address.

IP. Internet Protocol. A connectionless protocol that routes data through a network or interconnected networks. IP acts as an intermediary between the higher protocol layers and the physical layer.

IPSEC. Internet Protocol Security. A developing standard for security at the network or packet-processing layer of network communication.

ISP. Internet Service Provider

J

J2EE (Java 2 Enterprise Edition) The J2EE specification is based on the Java 2 platform, Standard Edition 1.3 and defines a standard for developing and deploying enterprise applications. The current version (v1.3) of the J2EE specification provides support for Enterprise JavaBeans (EJB), JavaServer Pages (JSP) and Java Servlet API component technologies as well as the Java Message Service (JMS), J2EE Connector Architecture (JCA), the Java API for XML Parsing (JAXP), and the Java Authentication and Authorization Service (JAAS).

JAR (Java Archive) A JAR file is a file that contains all the classes and associated files for a Java application, stored together in a single file in zip format. This provides for ease of use in deployment and also allows the contents to be compressed.

Java Application. A Java application is a program written in Java that executes locally on a computer. It allows programming operations in addition to those used in applets which can make the code platform dependent. It can access local files, create and accept general network connections, and call native C or C++ functions in machine-specific libraries.

JavaBeans. JavaBeans are Java components designed to be used on client systems. They are Java classes that conform to certain coding standards. They can be described in terms of their properties, methods and events. JavaBeans may be packaged with a special descriptor class called a BeanInfo class and special property editor classes in a JAR file. Java Beans may or may not be visual components. See `http://www.javasoft.com/beans/docs` for more information.

JDBC (Java Database Connectivity) JDBC is a Java API that allows Java programs to communicate with different database management systems in a platform-independent manner. Database vendors provide JDBC drivers for their platforms that implement the API for their database, allowing the Java developer to write applications to a consistent API no matter which database is used.

JDK (Java Development Kit) The set of development tools provided for use with the Java programming language. Now referred to as the *SDK* (software development kit).

JNDI (Java Naming and Directory Interface) An API that allows Java programs to interface and query naming and directory services in order to find information about network resources. JNDI is used in WebSphere to provide a directory of Enterprise Java Beans. See `http://java.sun.com/products/jndi/index.html` for more information..

JNI (Java Native Interface). Interface provided by the Java language to enable native (non-Java) code to be invoked.

JPDA (Java Platform Debugger Architecture) The distributed debugging architecture provided by the Java platform.

JSP (JavaServer Pages). HTML source files that include Java extensions to provide dynamic content and increased functionality. JavaServer Pages are compiled into Servlets before deployment. See `http://www.ibm.com/software/ebusiness/pm.html#JavaServer Pages`.

JVM (Java Virtual Machine). The virtual environment within which a Java application runs. The JVM provides for such features as platform neutrality and multi-threading.

L

LAN. Local Area Network. A computer network of devices connected within a limited geographical area for communication. A LAN can be connected to a larger network.

LDAP (Lightweight Directory Access Protocol) A software protocol enabling location of organizations, individuals, or other resources in a network. LDAP is a "lightweight" version of Directory Access Protocol (DAP), which is part of X.500, a standard for directory services in a network.

loopback adapter. See loopback interface.

loopback alias. An alternative IP address associated with the loopback interface. The alternative address has the useful side affect of not advertising on a real interface.

loopback interface. An interface that bypasses unnecessary communications functions when the information is addressed to an entity within the same system.

M

manager. One of several Dispatcher functions. The manager sets weights based on internal counters in the executor and feedback provided by the advisors. The executor then uses the weights to perform load balancing.erm3

mark down. To break all active connections to a server and stop any new connections or packets from being sent to that server.

mark up. To allow a server to receive new connections.

MEG. A plug-in encapsulates a correlated set of transcoding components (MEGs). A MEG is one of Monitor, Editor (Request or Document) or Generator. More information on MEGs and WBI (Web Intermediaries) can be found from `http://www.almaden.ibm.com/cs/wbi`.

Metric Server. Formerly known as Server Monitor Agent (SMA). Metric Server provides system-specific metrics to the Dispatcher manager.

metric. A process or command that returns a numeric value that can be used in load balancing on the network, for example, the number of users currently logged on.

MQSeries. A family of IBM products available for most platforms which provides an open, scalable, industrial-strength messaging and information system enabling enterprises and beyond to integrate business processes.

MRO (Multi Region Operation). Intercommunication between CICS regions in the same processor (or sysplex) without the use of SNA network facilities. This allows several CICS regions to communicate with each other, and to share resources such as files, terminals, temporary storage, and so on. Contrast with *intersystem communication.*

mutual high availability. Mutual high availability allows two Dispatcher machines to be both primary and backup for each other.

N

ndcontrol. Provides the interface to the Dispatcher component of Network Dispatcher.

ndserver. In Dispatcher, handles the requests from the command line to the executor, manager, and advisors.

netmask. For Internet subnetworking, a 32-bit mask used to identify the subnetwork address bits in the host portion of an IP address.

Network Address Port Translation (NAPT). also known as port mapping. This allows you to configure multiple server daemons within one physical server to listen on different port numbers.

Network Address Translation (NAT). A hardware device used to extend the usage of Internet addresses by allowing duplicate IP addresses to be used within a site while using unique addresses outside.

NIC. Network Interface Card. An adapter circuit board installed in a computer to provide a physical connection to a network.

O

OMG (Object Management Group). The consortium of software organizations that has defined the CORBA architecture.

ORB (**Object Request Broker).** A CORBA system component that acts as an intermediary between the client and server applications. Both client and server platforms require an ORB; each is tailored for a specific environment, but support common CORBA protocols and interfaces.

origin server. A server that hosts Web content, including applications. Origin servers typically are deployed within a firewall or intranet, with proxy servers placed closer to the edge of the network.

P

packet. The unit of data that is routed between an origin and a destination on the Internet or any other packet-switched network.

PDA. Personal Digital Assistant.

Persistence. Persistence is a term used to describe the storage of objects in a database to allow them to persist over time rather than being destroyed when the application containing them terminates. Enterprise Java Bean containers such as WebSphere provide persistence services for EJBs deployed within them.

ping. A command that sends Internet Control Message Protocol (ICMP) echo-request packets to a host, gateway, or router with the expectation of receiving a reply.

POP3. Post Office Protocol 3. A protocol used for exchanging network mail and accessing mailboxes.rm3

port. A number that identifies an abstracted communication device. Web servers use port 80 by default.

Preference Aggregation. Preference aggregation is the process where the client's device, network type, and any other information that might cause the transcoders to produce a different variant (or permutation) of the resource, are determined.

Preference Aggregator. Preference Aggregator is the process where the client's device, network type, and any other information that might cause the transcoders to produce a different variant (or permutation) of the resource, are determined.

Preference profiles. IBM WebSphere Transcoding Publisher uses preference profiles to represent the characteristics of devices and networks, and a default user profile to represent organizational policies. Each profile defines IBM WebSphere Transcoding Publisher how to treat documents that will be delivered to that device or over that network.

A preference profile can represent a particular type of device, such as a WorkPad, or a particular network type, such as a wireless network.

primary. In high availability for the Dispatcher, the primary is the machine that starts out actively routing packets. Its partner, the backup machine, monitors the status of the primary machine and takes over if necessary

priority. In rules-based load balancing, the level of importance placed upon any given rule. The Dispatcher evaluates rules from the first priority level to the last priority level.

private network. A separate network on which Dispatcher communicates with clustered servers for performance reasons.

protocol. The set of rules governing the operation of functional units of a communication system if communication is to take place. Protocols can determine low-level details of machine-to-machine interfaces, such as the order in which bits from a byte are sent; they can also determine high-level exchanges between application programs, such as file transfer.

Proxy. Transcoding Publisher connects through a proxy server that is configured with a firewall to manage network traffic and to protect your network from outside intrusion.

Q

Quality of Service. QoS. The performance properties of a network service, including throughput, transit delay, and priority. Some protocols allow packets or streams to include QoS requirements.

quiesce. To end a process by allowing operations to complete normally.

R

reach address. In high availability for the Dispatcher, the address of the target to which the advisor should issue pings to see if the target is responding.

reach. In Dispatcher, an advisor that issues pings to a given target and reports whether that target is responding.

return address. A unique IP address or host name. It is configured on the Dispatcher machine and used by Dispatcher as its source address when load balancing the client's request to the server.

reverse proxy. A network configuration in which a proxy server receives Internet requests and forwards them to one or more origin servers for reply. The address of the origin server is hidden. Also known as a surrogate configuration.

RMI. Remote Method Invocation (RMI) is a lightweight distributed object protocol that allows Java objects to call each other across a network. RMI is part of the core Java specification. See `http://java.sun.com/products/jdk/rmi/index.html` for more information.

Part of the Java programming language library that enables a Java program running on one computer to access the objects and methods of another Java program running on a different computer.

router. A device that forwards packets between networks. The forwarding decision is based on network layer information and routing tables, often constructed by routing products.

rule type. In rules-based load balancing, an indicator of the information that is evaluated to determine whether a rule is true.

rule. In rules-based load balancing, a mechanism for grouping servers such that a server can be chosen based on information other than the destination address and port.

S

Scalability. Scalability is an abstract attribute of software that refers to its ability to handle increased data throughput without modification. WebSphere handles scalability by allowing execution on a variety of hardware platforms that allow increased performance and clustering.

scalable. Pertaining to the capability of a system to adapt readily to a greater or lesser intensity of use, volume, or demand. For example, a scalable system can efficiently adapt to work with larger or smaller networks performing tasks of varying complexity.

server address. The unique code assigned to each computer that provides shared services to other computers over a network; for example, a file server, a print server, or a mail server. A standard IP address is a 32-bit address field. The server address can be either the dotted decimal IP address or the host name.

server. A computer that provides shared services to other computers over a network; for example, a file server, a print server, or a mail server. (The Dispatcher load balancing system groups multiple server machines into one virtual server and forwards requests to the cluster to one of the individual machines for processing.)

service. A function provided by one or more nodes; for example, HTTP, FTP, Telnet. shell. The software that accepts and processes command lines from a user's workstation. The Korn shell is one of several UNIX shells available.

Servlets. Servlets are Java classes that run on Web servers to provide dynamic HTML content to clients. They take as input the HTTP request from the client and output dynamically generated HTML. For more information on servlets see `http://www.ibm.com/software/ebusiness/pm.html#Servlets`.

session bean. An enterprise bean that is created by a client and that usually exists only for the duration of a single client/server session. A session bean performs operations such as calculations or accessing a database for a client. While a session bean may be transactional, it is not recoverable in the event of a system crash. Session beans can be stateful.

site name. A site name is an unresolvable host name that the client will request. For example, a Web site has 3 servers (1.2.3.4, 1.2.3.5, and 1.2.3.6) configured for site name www.dnsload.com. When a client requests this site name, one of the three server IP addresses will be returned as the resolution. The site name must be a fully qualified domain name, for example: dnsload.com. An unqualified name, for example, dnsload, is invalid for a site name.

SOCKS. A SOCKS server is a proxy server that uses a special protocol, sockets, to forward requests. Transcoding Publisher connects through a SOCKS server that is configured with a firewall to manage network traffic and to protect your network from outside intrusion (it supports Versions 4 and 5 SOCKS servers).

source address. In high availability for the Dispatcher, the address of the high availability partner machine that sends heartbeats.

SPARC. Scalable processor architecture.

SQL (Structured query language). A standard interactive and programming language for getting information from and updating a relational database. Queries take the form of a command language that lets you select, insert, update or find the location of data. Also an ANSI and an ISO standard.

SQLJ. A series of specifications for ways to use the Java programming language with SQL.

SSL. Secure Sockets Layer. A security scheme developed by Netscape Communications Corp. along with RSA Data Security Inc. SSL allows the client to authenticate the server and all data and requests to be encrypted. The URL of a secure server protected by SSL begins with https (rather than http).

stateful session bean. A type of session bean that has a client-specific conversational state, which it maintains across methods and transactions; for example a "shopping cart" object would maintain a list of the items selected for purchase. A stateful session bean that manages its own transactions can begin an OTS transaction in one method and commit or roll it back in a subsequent method. Contrast with *stateless session bean* See also *session bean*.

stateless session bean. A type of session bean that has no client-specific (nor any other kind of) non transient state; for example a "stock quotation" object might simply return current share prices. A stateless session bean that manages its own transactions and begins a transaction must commit (or roll back) the transaction in the same method in which it started it. Contrast with *stateful session bean*. See also *session bean*.

sticky time. The interval between the closing of one connection and the opening of a new connection during which a client is sent back to the same server used during the first connection. After the sticky time, the client may be sent to a server different from the first.

strategy. In high availability for the Dispatcher, a keyword for specifying how recovery takes place following the failure of the active machine.

subnet mask. For Internet subnetworking, a 32-bit mask used to identify the subnetwork address bits in the host portion of an IP address. surrogate configuration.

SYN. A control bit in the incoming segment, occupying one sequence number, used at the initiation of a connection to indicate where the sequence numbering will start.

T

TCP. Transmission Control Protocol. A communications protocol used on the Internet. TCP provides reliable host-to-host exchange of information. It uses IP as the underlying protocol.

TCP/IP listener. In CICS, the function provided by the CICS Sockets listener task (CSOL) that handles incoming HTTP and IIOP requests.

TCP/IP. Transmission Control Protocol/Internet Protocol. A suite of protocols designed to allow communication between networks regardless of the communication technologies used in each network.

TCP62. A communication protocol available when using the CICS Universal Client. It utilizes the AnyNet function of IBM's Communication Server to provide LU6.2 communication over TCP/IP (LU6.2/IP). It also provides for dynamic configuration of the client SNA node.

Terminal Servlet. A servlet provided by the CICS Transaction Gateway which provides turnkey Web browser access to CICS 3270-interface transactions. The servlet translates browser requests to CICS interactions, using the EPI Java classes to convert URL-encoded HTTP browser input to 3270-format data inbound, and to convert 3270-format transaction output to HMTL for return to the browser. These classes are not available on the OS/390 and the terminal servlet runs only on supported distributed platforms.

Text Transcoder. Text Transcoder is the transcoder that can modify elements of a text document based on device, network, and potentially user preference information. The primary use of this Text Transcoder is to modify Hypertext Markup Language (HTML) documents to remove unsupported elements, reduce space usage, replace features such as images or frames with links, and otherwise tailor documents to make them render more gracefully on constrained devices

timeout. The time interval allotted for an operation to occur.

TOS. Type of service. A 1-byte field in the IP header of the SYN packet.

transaction processing. A style of computing that supports interactive applications in which requests submitted by users are processed as soon as they are received. Results are returned to the requester in a relatively short period of time. A transaction processing system supervises the sharing of resources for processing multiple transactions at the same time.

transaction. (1) In CICS, a unit of processing consisting of one or more application programs initiated by a single request. A transaction can require the initiation of one or more tasks for its execution. (2) A unit of work that transforms one or more resources from one consistent state to another consistent state. See also *Logical Unit of Work*.

Transcoder. Transcoder is a program that modifies the content of a document.

Transcoding. Transcoding is a technology that gives you the ability to make Web based information available on handheld and other new type devices economically and efficiently or on the slow network connections like a dial up modem connection. With transcoding, users receive information (text and images) tailored to the capabilities of the devices they are using and also tailored according to the capacity of the network being used.

It is also the process where the MEGs related to modifying the request, generating the original resource and all of the document (or resource) editing (or transcoding) occurs.

U

UDP. User Datagram Protocol. In the Internet suite of protocols, a protocol that provides unreliable, connectionless datagram service. It enables an application program on one machine or process to send a datagram to an application program on another machine or process. UDP uses the Internet Protocol (IP) to deliver datagrams.

UNIX System Services. The UNIX environment provided by the OS/390 operating system. Previously referred to as *OpenEdition*.

URL (Uniform Resource Locator). Format used to describe the resources located on the World Wide Web. A URL is of the form `protocol://server/pathname/document` It consists of the document's name preceded by pathname where the document can be found, the Internet domain name of the server that hosts the document and the protocol (such as http or ftp) by which the Web browser can retrieve the document from the server.

A standard way of specifying the location of an object, typically a Web page, on the Internet. URLs are the form of address used on the World-Wide Web. They are used in HTML documents to specify the target of a hyperlink, which is often another HTML document (possibly stored on another computer).

V

VAJ (VisualAge for Java). An IBM program product that provides an integrated development environment for Java development and testing.

VPN. Virtual Private Network. A network comprised of one or more secure IP tunnels connecting two or more networks.

W

WAP. Wireless Application Protocol.

Web Application Servers. A Web Application Server is a software program designed to manage applications at the second-tier of three-tier computing, that is, the business logic components. A Web Application Server manages applications that use data from back-end systems, such as databases and transaction systems, and provides output to a Web browser on a client. For more information see
 `http://www.ibm.com/software/ebusiness/appsrvsw.html`

Web. The network of HTTP servers that contain programs and files, many of them hypertext documents that contain links to other documents on HTTP servers. Also World Wide Web.

WebSphere Application Server. A program product which provides a long-running server process to run Web server programs written in Java. When used on OS/390, WebSphere Application Server provides a Web server — the IBM HTTP Server (known previously as Lotus Domino Go Webserver and the Internet Connection Secure Server). When used on distributed platforms it plugs in to a variety of vendor specific Web servers including the IBM HTTP Server (based on the Apache HTTP Server).

wizard. A dialog within an application that uses step-by-step instructions to guide a user through a specific task.

WTE. Web Traffic Express. An IBM caching proxy. Referred to as Caching Proxy in later releases of WebSphere Edge Server.

WTP. WebSphere Transcoding Publisher.

XML. XML, or eXtensible Markup Language, is a platform-independent and application-independent way of describing data using tags. XML (a subset of SGML) is similar to HTML in that it uses tags to describe document elements but different in that the tags describe the structure of the data rather than how the data is to be presented to a client. XML has the facility to allow data providers to define new tags as needed to better describe the data domain being represented. For more information see `http://www.ibm.com/developer/xml`.

XSL Stylesheets. XSL stylesheets are documents that describe a mapping between XML documents and visual data that can be presented to a client in a browser. XSL was a draft standard when this book was being written. The draft can be found at `http://www.w3.org/TR/WD-xs.`

Abbreviations and acronyms

ACL	access control list
ACPI	Advanced Configuration and Power Interface
ADO	ActiveX Data Object
AES	Advanced Encryption Standard
AFP	Advanced Function Printer
AFS	Andrew File System
AIX	Advanced Interactive eXecutive
AOR	Application owning region
APAR	authorized program analysis report
API	application programming interface
APPC	Advanced Program-to-Program Communication
APPN	Advanced Peer-to-Peer Network
ARP	Address Resolution Protocol
AS/IMS	Application Support/IMS
asate	application service at the edge
ASCII	American National Standard Code for Information Interchange
ATI	automatic transaction initiation
ATM	Asynchronous Transfer Mode
AWT	Abstract Windowing Toolkit
BIND	Berkeley Internet Name Domain
BMP	Batch Message Program, Bean Managed Persistence
BMS	Basic Mapping Support
CA	Certificate Authority
CAB	cabinet
CAE	Client Application Enabler
CAPI	cryptographic application programming interface
CARP	Cache Array Routing Protocol
CB	Component Broker
CBPDO	Custom Built Product Delivery Offering
CBR	Content Based Routing
CCF	Common Connector Framework
CCI	Common Client Interface
CD	compact disc
CDC	Content Distribution Client
CDF	Content Distribution Facility, Content Distribution Framework
CDMF	Commercial Data Masking Facility
CDN	Content Distribution Network
CDS	Content Distribution System
CECI	Command Level Interpreter Transaction
CGI	Common Gateway Interface
CICS	Customer Information Control System
CLI	Call Level Interface
COMMAREA	communications area
CORBA	Common Object Request Broker Architecture
CSD	corrective service delivery, corrective service distribution, corrective service diskette

CSI	Consolidated Software Inventory
CSS	Cascading Style Sheet
CTG	CICS Transaction Gateway
CWS	CICS Web support
DASD	Direct Access Storage Device
DB2LSX	DB2 LotusScript Extension
DBCS	double byte character set
DCAR	Digital Certificate Access Requester
DCAS	Digital Certificate Access Server
DCE	Distributed Computing Environment
DDCS	Data Definition Control Support
DDE	dynamic data exchange
DDF	Distributed Data Facility
DEC	Digital Equipment Corporation
DECS	Domino Enterprise Connection Services
DES	Data Encryption Standard
DIT	Directory Information Tree
DLUR	dependent LU requestor
DMT	Directory management tool
DMZ	demilitarized zone
DN	distinguished name
DNS	Domain Name System
DOM	Document Object Model
DPI	dots per inch
DPL	Distributed Program Link
DRDA	Distributed Relational Database Architecture
DSCP	Differentiated Services Control Point
DTP	Distributed Transaction Processing
EAB	Enterprise Access Builder
EAR	Enterprise Archive file
EBCDIC	Extended Binary Coded Decimal Interchange Code
ECA	External Cache Adapter
ECI	External Call Interface
ECLPS	Emulator Class Library Presentation Services
EHLLAPI	enhanced high level language application programming interface
EJB	Enterprise Java Beans
EJS	Enterprise Java Services
ELF	express logon function
ELP	enhanced local preferences
EPI	External Presentation Interface
ESA	Enterprise Systems Architecture
ESI	Edge side includes, External Security Interface
ESP	Encapsulating Security Payload
EXCI	External CICS Interface
FCRA	Fast Cache Response Accelerator
FDDI	Fiber Distributed Data Interface
FEPI	Front End Programming Interface
FTP	File Transfer Protocol
GC	garbage collector
GDI	graphical device interface
GEM	Global Enterprise Manager
GID	Group Identification Number
GIF	Graphics Interchange Format
GRE	Generic Routing Encapsulation
GSK	Global Security Toolkit

GUI	graphical user interface
GWAPI	Go Webserver API
HABJ	Host Access Beans for Java
HACL	Host Access Class Library
HACLJ	Host Access Class Library for Java
HACMP	High Availability Cluster Multiprocessing
HFS	Hierarchical File System
HLLAPI	high level language application programming interface
HMAC	Hashed Message Authentication Code
HOD	Host On-Demand
HPR	high performance routing
HPT	host print transform
HP-UX	Hewlett-Packard UNIX
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IBM	International Business Machines Corporation
ICAPI	Internet Connection Application Programming Interface
ICP	Internet Caching Protocol
ICU	International Component for Unicode
IDE	Integrated Development Environment
IDEA	International Data Encryption Algorithm
IDL	Interface Definition Language
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IHS	IBM HTTP Server
IIOP	Internet Inter-ORB Protocol
IMAP	Internet Mail Access Protocol
IMS	Information Management System
IMS/ESA	IMS/Enterprise Systems Architecture
IP	Internet Protocol
IPSec	IP Security Architecture
ISAKMP	Internet Security Association and Key Management Protocol
ISC	Inter-System Communication
ISO	International Organization for Standardization
ISP	Internet Service Provider
ISS	Interactive Session Support
ITOC	IMS TCP/IP OTMA Connection
ITSO	International Technical Support Organization
IWT	IMS Web Templates
J2EE	Java 2 Enterprise Edition
J2ME	Java 2 Micro Edition
J2SE	Java 2 Standard Edition
JAR	Java Archive
JCA	Java Connector Architecture
JCL	job control language
JDBC	Java Database Connectivity
JDK	Java Development Kit
JITOC	Java IMS TCP/IP OTMA Connection Connector
JNDI	Java Naming and Directory Interface
JNI	Java Native Interface
JPEG	Joint Photographic Experts Group
JRE	Java Runtime Environment
JSP	JavaServer Pages
JTA	Java Transaction API

JVM	Java Virtual Machine
L2TP	Layer 2 Tunnel Protocol
LAN	local area network
LC LSX	Lotus Connector LotusScript Extension
LDAP	Lightweight Directory Access Protocol
LDIF	Lightweight Directory Interface File
LEI	Lotus Enterprise Integrator
LLC2	link level control 2
LPAR	Logical partition mode
LSX	LotusScript Extension
LTPA	Lightweight third party authentication
LU	logical unit
MAC	Media Access Control, Message Authentication Code
Mbps	megabits per second
MBps	megabytes per second
MCDS	Master Content Distribution Server
MIB	management information base
MIME	Multipurpose Internet Mail Extensions
MPA	multi-protocol adapter
MPP	Message Processing Program
MQ	Message and Queueing
MQEI	MQSeries Enterprise Integrator
MQI	Message and Queueing Interface
MQLSX	MQSeries Link LotusScript Extension
MQM	Message Queue Manager
MVS	Multiple Virtual Storage
NAP	network access points
NAT	Network Address Translation
NCSA	National Center for Supercomputing Applications
ND	Network Dispatcher
NDIS	network driver interface specification
NFA	Non-forwarding address
NIC	Network Interface Card
NLS	National Language Support
NNTP	NetNews transfer protocol
NPT	non-programmable terminal
ODBC	Open Database Connectivity
ODG	object dependency graph
OHIO	Open Host Interface Objects
OIA	operator information area
OLTP	On-Line Transaction Processing
OMG	Object Management Group
ORB	Object Request Broker
OS/2	Operating System/2
OSA	Open Systems Adapter
OSE	open servlet engine
OSF	Open Software Foundation
OTMA	Open Transaction Manager Access
PAC	proxy automatic configuration
PCL	printer control language
PCOMM	IBM Personal Communications
PDF	printer definition file
PDT	printer definition table
PGP	Pretty Good Privacy
PICS	Platform for Internet Content Selection
PKCS	Public Key Cryptographic Standard
PKI	Public Key Infrastructure

POP	points of presence, Protected object policy
POP3	Post Office Protocol 3
PPDS	personal printer definition stream
PPP	Point-to-Point Protocol
PSP	Preventive Service Planning
PTF	Program Temporary Fix
PWS	programmable workstation
RACF	Resource Access Control Facility
RCA	Remote Cache Access
RDBMS	Relational Database Management System
REXX	Restructured Extended eXecutor Language
RFC	Request for Comment
RIO	Remote Integration Object
RMI	Remote Method Invocation
RPC	Remote Procedure Call
RSACI	Recreational Software Advisory Council on the Internet
RTSP	Real Time Streaming Protocol
SAX	Simple API for XML
SCCI	Screen Customizer Component Interface
SCS	SNA character string
SDA	Server Directed Affinity
SDK	Software Development Kit
SDLC	synchronous data link control
SET	Secure Electronic Transactions
SGML	Standard Generalized Markup Language
SLA	Service Level Agreement
SLP	service location protocol
SMA	Server Monitor Agent
SMP/E	System Modification Program/Extended
SMS	Systems Management Server
SMTP	Simple Mail Transfer Protocol
SNA	Systems Network Architecture
SNMP	Simple Network Management Protocol
SPARC	Scalable processor architecture
SQL	Structured Query Language
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TEC	Tivoli Enterprise Console
TIF	Tagged Image Format
TM	Transaction Manager, Trigger Monitor
TOS	Type Of Service
TQoS	Transactional Quality of Service
TSO	Time Sharing Option
UDC	user-defined characters
UDP	User Datagram Protocol
UID	User Identification Number
URI	Universal Resource Identifier
URL	Universal Resource Locator, Uniform Resource Locator
USB	universal serial bus
USSMSG10	Unformatted Systems Services, message number 10
VM	virtual machine
VPN	Virtual Private Network
VT	virtual terminal
VTAM	Virtual Telecommunication Access Method

W3C	World Wide Web Consortium
WAN	Wide Area Network
WAND	Wide Area Network Dispatcher
WAP	Wireless Application Protocol
WAR	Web module Archive
WAS	WebSphere Application Server
WAS AE	WebSphere Application Server Advanced Edition
WAS AEs	WebSphere Application Server Advanced Single Server Edition
WCDP	Web Content Distribution Protocol
WCPP	Web Content Publishing Protocol
WLM	workload manager
WML	Wireless Markup Language
WPAD	Web Proxy Auto Discovery
WSES	WebSphere Edge Server
WTE	Web Traffic Express
WWW	World Wide Web
XCF	Cross-System Coupling Facility
XLGW	XML Legacy Gateway
XML	eXtensible Markup Language
XSL	eXtensible Style Language

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information on ordering these publications, see "How to get IBM Redbooks" on page 396.

- *WebSphere Edge Server: Working with Web Traffic Express and Network Dispatcher,* SG24-6172
- *IBM WebSphere Edge Server: New Features and Functions in Version 2,* SG24-6511
- *WebSphere V4 Advanced Edition Handbook,* SG24-6176
- *Self-Service Patterns using WebSphere Application Server V4.0,* SG24-6175
- *Self-Service Applications using IBM WebSphere V4.0 and IBM MQSeries Integrator,* SG24-6160
- *Application Integration Pattern with Domino and WebSphere,* SG24-6255

Other resources

These publications are also relevant as further information sources:

- *Patterns for e-business: A Strategy for Reuse*, by Jonathan Adams, Srinivas Koushik, Guru Vasudeva, George Galambos, published by IBM Press, ISBN: 1-931182-02-7
- *WebSphere Edge Server for Multiplatforms Getting Started Version 2,* SC09-4566
- *WebSphere Edge Server for Multiplatforms Administration Guide Version 2,* GC09-4567
- *WebSphere Edge Server for Network Dispatcher Administration Guide Version 2,* GC31-8496

- *Tivoli Access Manager Plug-in for Edge Server Administration Guide Version 3.9,* GC32-4685
- *Tivoli Access Manager Base Installation Guide Version 3.9,* GC32-0844
- *Tivoli Access Manager Base Administration Guide Version 3.9,* GC23-4684

Referenced Web sites

These Web sites are also relevant as further information sources:

- IBM Patterns for e-business and the PDK sample application

  `http://www.ibm.com/developer/patterns`
- WebSphere Edge Server product overview

  `http://www-3.ibm.com/software/webservers/edgeserver/`
- SecureWay products

  `http://www.tivoli.com/security/`

How to get IBM Redbooks

You can order hardcopy Redbooks, as well as view, download, or search for Redbooks at the following Web site:

ibm.com/redbooks

You can also download additional materials (code samples or diskette/CD-ROM images) from that site.

IBM Redbooks collections

Redbooks are also available on CD-ROMs. Click the CD-ROMs button on the Redbooks Web site for information about all the CD-ROMs offered, as well as updates and formats.

Index

IBM

Redbooks

Patterns for the Edge of Network

(0.5” spine)
0.475”<->0.875”
250 <-> 459 pages

Patterns for the Edge of Network

Runtime Patterns with WebSphere Edge Server

High availability guidelines

High performance guidelines

Patterns for e-business are a group of proven, reusable assets that can help speed the process of developing applications.

In this IBM Redbook, we describe guidelines and options for the selection of Runtime patterns that include high availability and high performance considerations in the design process. Specifically, we address software and node configuration scenarios within the demilitarized zone (DMZ) and give examples of implementation procedures.

Part 1 of the redbook provides you with an overview of Patterns for e-business. It details Runtime patterns for high availability and high performance. It then gives you possible product mappings for implementation of the chosen runtime pattern.
Part 2 is a set of guidelines for selecting your Runtime pattern within the DMZ. It includes information on technology options, high availability and high performance definitions, and a review of security considerations.
Part 3 takes you through installation and working examples, showing the implementation of high availability and high performance features of WebSphere Edge Server.

SG24-6822-00 ISBN0738426121