Overview Guide

Netscape Application Server

Version 4.0

Recycled and Recyclable Paper

Contents

Preface

The *Overview Guide* provides background information about the product components, features, and system architecture of Netscape Application Server (NAS).

The *Overview Guide* is only one of several documents that help you develop, deploy, and manage web-based enterprise applications. The following table lists the tasks and concepts that are described in the Netscape Application Server (NAS) and Netscape Application Builder (NAB) printed manuals and online readme file. If you are trying to accomplish a specific task or learn more about a specific concept, refer to the appropriate manual.

Note that the printed manuals are also available as online files in PDF and HTML format.

For information about	See the following	Shipped with
Late-breaking information about the software and the documentation	`readme.htm`	NAS 4.0, NAS 4.0 Developer Edition (Solaris), NAB 4.0
Installing Netscape Application Server and its various components (Web Connector plug-in, Netscape Application Server Administrator), and configuring the sample applications	Installation Guide	NAS 4.0 Developer Edition (Solaris), NAS 4.0
Installing Netscape Application Builder	`install.htm`	NAB 4.0
Basic features of NAS, such as its software components, general capabilities, and system architecture	Overview	NAS 4.0, NAS 4.0 Developer Edition (Solaris), NAB 4.0

For information about	See the following	Shipped with
Deploying Netscape Application Server at your site, by performing the following tasks: • Planning your Netscape Application Server environment • Integrating the product within your existing enterprise and network topology • Developing server capacity and performance goals • Running stress tests to measure server performance • Fine-tuning the server to improve performance	Deployment Guide	NAS 4.0
Administering one or more application servers using the Netscape Application Server Administrator tool to perform the following tasks: • Deploying applications with the Deployment Manager tool • Monitoring and logging server activity • Setting up users and groups • Administering database connectivity • Administering transactions • Load balancing servers • Managing distributed data synchronization	Administration Guide	NAS 4.0
Migrating your applications to the new Netscape Application Server 4.0 programming model from version 2.1, including a sample migration of an Online Bank application provided with Netscape Application Server	Migration Guide	NAS 4.0, NAS 4.0 Developer Edition (Solaris), NAB 4.0

For information about	See the following	Shipped with
Creating NAS 4.0 applications within an integrated development environment by performing the following tasks: • Creating and managing projects • Using wizards • Creating data-access logic • Creating presentation logic and layout • Creating business logic • Compiling, testing, and debugging applications • Deploying and downloading applications • Working with source control • Using third-party tools	User's Guide	NAB 4.0
Creating NAS 4.0 applications that follow the new open Java standards model (Servlets, EJBs, JSPs, and JDBC), by performing the following tasks: • Creating the presentation and execution layers of an application • Placing discrete pieces of business logic and entities into Enterprise Java Beans (EJB) components • Using JDBC to communicate with databases • Using iterative testing, debugging, and application fine-tuning procedures to generate applications that execute correctly and quickly	Programmer's Guide (Java)	NAS 4.0 Developer Edition (Solaris), NAB 4.0
Using the public classes and interfaces, and their methods in the Netscape Application Server class library to write Java applications	Server Foundation Class Reference (Java)	NAS 4.0 Developer Edition (Solaris), NAB 4.0

For information about	See the following	Shipped with
Creating NAS C++ applications using the NAS class library by performing the following tasks: • Designing applications • Writing AppLogics. • Creating HTML templates • Creating queries • Running and debugging applications	Programmer's Guide (C++)	Order separately
Using the public classes and interfaces, and their methods in the Netscape Application Server class library to write C++ applications	Server Foundation Class Reference (C++)	Order separately

1

Product Overview

This chapter explains the role of Netscape Application Server (NAS) within the multitier enterprise environment. In addition, the chapter summarizes the main product components that come with NAS.

This chapter contains the following sections:

- The Multitiered Environment

- Product Components

The Multitiered Environment

Netscape Application Server is the middleware between enterprise data sources and the clients that access those data sources. Business code is stored and processed on Netscape Application Server (NAS) rather than on clients. An application is deployed and managed in a single location, and the application is accessible to large numbers of heterogeneous clients.

NAS applications run in a distributed, multitiered environment. This means that an enterprise system might consist of several application servers (computers running the Netscape Application Server software), along with multiple database servers and web servers. Your application code can be distributed among the application servers.

Overall, the machines and software involved are divided into three tiers:

- a client tier, represented by web browsers or rich clients (such as a Java application).

- a server tier, represented by a web server (such as Netscape Enterprise Server) and an application server (such as Netscape Application Server).

- a data tier, represented by relational databases or other back-end data sources.

The following figure shows a multitiered environment:
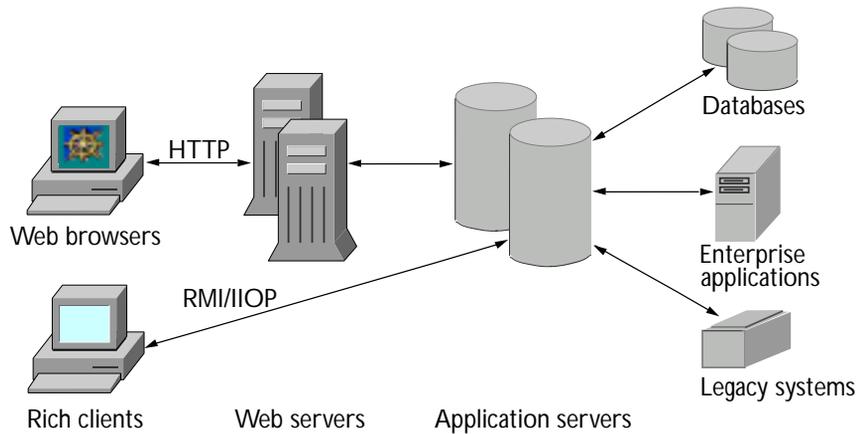
End users interact with client software, typically a web browser, to use the application.

- When a request originates from a web browser, it is sent to the web server. Assuming the request requires application processing or data access, the web server forwards the request to Netscape Application Server (NAS).

- When a request originates from a CORBA client, it is sent to NAS by way of an RMI/IIOP link. (An RMI/IIOP-based product is expected to ship after NAS 4.0.)

NAS handles requests by running the appropriate application code (and accessing data sources if needed). NAS returns the results to the web server, which in turn forwards the reply back to the client.

For more information about the components and interactions in the multitier environment, see the *Programmer's Guide*.

Product Components

This section describes the software and product components of NAS. This section consists of the following topics:

- Programming APIs
- System Services and Application Services
- Sample Applications
- NAS Administrator
- NAS Deployment Manager
- Netscape Directory Server

Programming APIs

NAS supports several industry-standard Java APIs. In particular, NAS supports the APIs and technologies as defined by the following specifications:

- *Java Servlet 2.1 API Specification*
- *Enterprise JavaBeans 1.0 Specification*
- *JavaServer Pages 0.92 Specification*
- *JDBC 2.0 Core API Specification*
- *JDBC 2.0 Standard Extensions Specification*

Note that NAS 4.0 provides full support for JDBC 1.0 and partial support for JDBC 2.0. The supported JDBC 2.0 functionality is described in the *Programmer's Guide (Java)*.

For building application components written in C++, NAS provides the Foundation Class Library. Java application developers can also use the NAS Foundation Class Library to leverage additional capabilities of NAS that are not supported through standard APIs.

For more information about the NAS Foundation Class Library, see the *Programmer's Guide* or the *Netscape Application Server Foundation Class Reference*. Both of these documents are available in Java or C++ versions.

For more information about the industry-standard Java APIs that are supported in NAS 4.0, see the appropriate API specification. All specifications are accessible from *installdir*/nas/docs/index.htm, where *installdir* is the location in which you installed NAS.

System Services and Application Services

System and application services provide a variety of application-level capabilities and system-level capabilities. These services enable the development, deployment, and management of complex business-logic and transaction-based applications.

For more information about these services, see "System Components" in Chapter 3, "Netscape Application Server Architecture."

Sample Applications

NAS includes sample web-based applications, enabling you to more quickly learn techniques for developing and deploying applications in a NAS environment.

One sample presents a bookstore application that simulates browsing, searching, and ordering books online. This Java application demonstrates the NAS application model that uses industry-standard components such as servlets, JavaServer Pages, Enterprise JavaBeans, and data access with JDBC. For information about installing or using the online bookstore application, see the *Installation Guide* or the *Programmer's Guide (Java)*.

Another sample presents a banking application that simulates a user session with an online account. This sample demonstrates techniques for migrating existing applications to comply with the industry-standard Java application model. For information about installing the bank application, see the *Installation Guide*. For details about the application code, see the *Migration Guide*.

NAS Administrator

NAS Administrator is a GUI tool that contains several smaller tools for managing one or more NAS machines or applications. For more information, see "Management Capabilities" in Chapter 2, "Features." For detailed information about using NAS Administrator, see the *Administration Guide*.

NAS Deployment Manager

An application must be deployed before it can be used, and NAS Deployment Manager is a GUI tool that makes application deployment easier. You access this tool either from NAS Administrator or from Netscape Application Builder.

When you deploy an application, NAS Deployment Manager installs all the application's files and registers all of its components on the destination server (a server on which NAS has been installed).

For information about using NAS Deployment Manager to deploy applications, see the *Administration Guide*.

Netscape Directory Server

Netscape Directory Server provides a comprehensive, enterprise-wide directory service for managing information about users, groups, and access control lists. NAS 4.0 includes Netscape Directory Server 4.0, which supports versions 2 and 3 of the Lightweight Directory Access Protocol (LDAP).

For more information, see the NAS *Deployment Guide*. In addition, consult the *Installation Guide* for Netscape Directory Server 4.0.

Product Components

Features

This chapter describes the key features of Netscape Application Server.

This chapter contains the following sections:

- Rapid Application Development

- High Scalability

- High Performance

- High Availability

- Session and State Management

- Security

- Management Capabilities

Rapid Application Development

NAS enables rapid development of enterprise applications through the following features, which are described in this section:

- a multi-tier application model.
- core services for building high-performance, scalable, transactional applications.
- support for a variety of application development tools.

Application Model

An application model is the conceptual division of a software application into functional components. The NAS application model promotes code reusability and faster application deployment.

The NAS application model divides an application into multiple layers: presentation, business logic, and data access. Presentation is further separated to distinguish page layout and presentation logic. Data access refers to both databases and other data sources.

The NAS application model is component-oriented. For Java applications, the application model has been enhanced. The model now incorporates standard components based on Java technologies.

The following table lists the main components that make up the functions of an application in the NAS environment:

Functionality in Application Model	Application Components	
	Java Applications	C++ Applications
Presentation logic	servlets	AppLogics
Page layout	JavaServer Pages	HTML templates
Business logic	Enterprise JavaBeans	AppLogics
Database access	Enterprise JavaBeans using JDBC; query files	AppLogics using the NAS API; query files
Access to other data sources	Extensions	Extensions

These application component fall into two categories:

- Industry-Standard Components
- Other Components

Industry-Standard Components

For developing Java applications, we recommend that you use standard components whenever possible. These standards-based components include servlets, JavaServer Pages (JSPs), and Enterprise JavaBeans (EJBs), described as follows:

- Servlets are Java classes that define page logic and page navigation. Servlets also support the creation or invocation of business components.

- JSPs are web browser pages written in a combination of HTML, JSP tags, and Java.

- EJBs encapsulate an application's business rules and business entities.

In addition, application components can invoke JDBC calls. JDBC is a standard API for database connectivity.

For more information about using these standard components in NAS applications, see the *Programmer's Guide (Java)*.

Other Components

Other application components include AppLogics, HTML templates, query files, and extensions. An application must use these components if it is written in C++ or if it will run in the NAS 2.x environment. And if your application must access propriety enterprise data sources, you may need to use prebuilt extensions or create your own.

- An AppLogic is a set of programming instructions, written in C++ or Java, that perform a well-defined, modular task within an application.

- HTML templates are text files that merge HTML with dynamic data to produce formatted output. Templates use special markup tags called GX tags.

- Query files are text files that contain SQL commands, such as for querying or updating a database.

- Extensions enable business logic components to connect to custom or proprietary enterprise resources. Extensions are persistent modules, written in C++ or Java, that are dynamically loaded into NAS and are accessed by multiple AppLogics or EJBs over the life of the extension.

  — Ready-to-use extensions are provided as part of NAS Integration Solutions. NAS Integration Solutions are available for MQSeries, Tuxedo, CICS, IMS, and R/3 applications. These applications can be easily integrated with applications deployed on NAS.

  — Developers can create their own custom extensions using Netscape Extension Builder, a separately packaged, GUI-based tool that integrates with NAS.

For more information about AppLogics, HTML templates, and query files, see the *Programmer's Guide (C++)* or the *Migration Guide*. For more information about extensions, consult your Netscape Extension Builder documentation.

Core Services

NAS provides a set of application services and system services for building, deploying, and managing high-performance, scalable, transactional applications. These services include built-in state and session management, request and transaction management, results caching, connection caching, and so on.

For more information about these services, see "System Components" in Chapter 3, "Netscape Application Server Architecture."

A Choice of Tools

Netscape Application Server supports applications written in either Java or C++. However, Java applications are easier to develop and maintain, because they can take advantage of the enhanced, standards-based application model.

Application developers can choose from among various tools to build applications. These tools can range from simple text editors, to visual Java editors and visual HTML editors, to integrated development environments (IDEs).

Netscape's tools include Netscape Application Builder and Netscape Extension Builder. These tools are tightly integrated with NAS but are packaged separately.

- Netscape Application Builder is used for building applications and deploying them on NAS. Netscape Application Builder also interoperates with third-party tools such as Symantec Visual Cafe and Macromedia Dreamweaver.

- Netscape Extension Builder is used for designing and building custom extensions.

High Scalability

Netscape Application Server has a scalable architecture. This means that applications can be built to meet the needs of initial deployment. Applications can then be scaled as business needs grow.

Application scaling is accomplished in two main ways: either by adding more servers to a cluster of servers, or by adding more CPUs to a multi-CPU system. Application logic can then be deployed to the new servers. Developers do not need to change any application logic as the user base grows.

In addition, application tasks can be assigned to the server best able to process the request efficiently. This is accomplished either through application partitioning or through dynamic load balancing.

Application Partitioning

The Netscape Application Server architecture supports application partitioning, which allows logic to be distributed across servers as an application scales to accommodate heavier loads. Using Netscape Application Server Administrator, system administrators can partition an application into functional areas.

For example, in an online catalog application, the application logic for order processing, inventory management, and checkout processing can reside on different servers. Regardless of how applications are partitioned and distributed, the application functions as a single, cohesive unit.

Application logic can also be grouped where each group consists of related operations. For example, a group might contain all logic associated with order processing. Each application component can belong to one or more groups. Applications can also share application logic.

System administrators can deploy these groups of application logic objects locally or globally across application servers in the following ways:

- Portions of an application might uniquely reside on different Netscape Application Servers, yet still run as a single application. In this way, application logic can be stored on the server that can run it most efficiently. For example, data-intensive application logic can be run on the server that is closest to the data source to avoid latencies associated with accessing remotely located data.

- For load-balanced applications, the same group of application logic objects can be stored on multiple servers. This allows an application to run the application logic more efficiently on the server with the most available resources.

- Applications might dynamically share certain application logic objects. For example, all applications in a network might share the same application logic for user login and authentication, or for credit card authorization.

Application partitioning gives system administrators tremendous flexibility to scale and tune the performance of applications. In addition, having application components stored on multiple servers helps ensure high application availability in the event of a server shutdown.

Dynamic Load Balancing

Netscape Application Server supports dynamic load balancing to provide optimal performance levels under heavy loads. With load balancing enabled, the Netscape Application Server can direct certain requests to be run on an available server instead of waiting for a busy server to become available. If one server is overburdened, another can take its place to process the request. The Load Monitor tabulates information on server resource availability.

To use load balancing, application logic must be partitioned across all Netscape Application Servers that might process the application logic at run time. System administrators must determine if an entire application or specific parts of an application should be load balanced.

For optimal performance and resource utilization, system administrators can deploy application logic on servers that are best configured to handle execution of that logic. Each Netscape Application Server has its own load balancing module which determines the optimal server to process an incoming request.

Partitioning characteristics are dynamically known to all servers in the cluster. Netscape Application Servers regularly update their load statistics and broadcast them to other Netscape Application Servers in the cluster. Based on load balancing factors, requests are dynamically routed to servers. Load balancing factors are configured using Netscape Application Server Administrator.

High Performance

Netscape Application Server is a high performance, multi-threaded, and multi-processing application server. NAS can handle a high number of concurrent requests, database connections, and sessions, and provides high performance even under heavy loads.

NAS delivers high performance between web servers, other NAS machines, and heterogeneous back-end data sources through the following features:

- Database Connection Caching

- Result Caching

- Data Streaming

- Multi-threaded Capabilities

- Optimized Communication with Web Servers

Aside from the application server, other factors affecting application performance include network topology, network and server hardware, database architecture, and application programming.

Database Connection Caching

To improve performance, the Netscape Application Server caches database connections so that commonly used, existing connections are re-used rather than re-established each time. Connection caching avoids the overhead involved in creating a new database connection for each request.

Application developers can enable database connection caching in their application logic. At run time, when a request creates a new connection to a database, Netscape Application Server stores the connection in the cache. When the request has completed using the connection, the connection is marked as free in the cache. If a new request is made to the same database by the same user, Netscape Application Server can first check the cache and use an existing free connection rather than creating a new one. If the freed connection remains unused after a specified time-out period, it is released by the server.

System administrators can use the Netscape Application Server Administrator to specify server-wide settings for database connection caching, such as the initial number of slots in the cache and the time-out limit for free connections, and so on.

Using Netscape Application Server Administrator, system administrators can monitor server performance and tune the number of available connections in the cache. This ensures an optimal ratio of cached connections to system resources.

Result Caching

Netscape Application Server improves application performance by caching the results of application logic execution. Developers can optionally enable this feature in their applications.

If caching is enabled, Netscape Application Server saves the application logic's input parameters and results in the cache. The next time the Netscape Application Server executes the same request, the server can first check the cache to determine whether the input parameters match the cached input parameters. If they match, the server retrieves the results from the cache instead

of executing the request again. Result caching is especially effective for large data requests that involve lengthy processing time and for frequently accessed application logic.

Data Streaming

Netscape Application Server provides data streaming facilities. Streaming improves performance by allowing users to begin viewing results of requests sooner, rather than waiting until the complete operation has been processed. Application developers can explicitly control what data is streamed, or allow the system to provide automatic streaming.

Streaming is especially useful for large data sets involving lengthy queries. For example, suppose a user requests a price list containing 10,000 items. The application can process the query and display items to the user as they become available, for example, 40 at a time (or one full page view), rather than wait until all 10,000 items have been retrieved from the database.

Multi-threaded Capabilities

Netscape Application Server supports the multi-threading capabilities of the host operating system. An application can optimize performance by processing requests on multiple threads, which maximizes CPU resource utilization.

Application developers automatically take advantage of multi-threading in their applications. In addition, developers can run database operations such as queries, inserts, updates, deletes, and so on, asynchronously. Asynchronous operations allow an application to do other work while a time-consuming operation, such as a large query, runs in the background.

System administrators can use Netscape Application Server Administrator tool to specify settings for multi-threading, such as the following:

- minimum and maximum number of threads to handle all requests

- minimum and maximum number of threads to handle asynchronous database requests.

Typically, administrators will monitor server performance and tune the number of available threads to achieve an optimal ratio of threads to system resources.

Optimized Communication with Web Servers

Netscape Application Server optimizes application performance through tighter integration with web servers. This integration occurs using Web Connector Plug-ins and corresponding Listeners. Netscape Application Server supports NSAPI, ISAPI, and optimized CGI for Netscape, Microsoft, and CGI-compatible Web servers, respectively.

High Availability

Many enterprise applications must be accessible (available) to users 24 hours a day, 7 days a week. Netscape Application Server provides a highly available and reliable solution through the use of load balancing and dynamic failover (also called failure recovery).

NAS enables you to distribute all or part of an application across multiple servers. As a result, if one server goes down, the other servers can continue to handle requests.

NAS minimizes downtime by providing automatic application restarting. In addition, NAS maintains and replicates distributed user-session information and distributed application-state information. Information is maintained as long as more than one NAS installation is running in a cluster with the server that crashed.

Developers need not be concerned with building recovery and scalability features into their application. The application inherits these features simply by being hosted on the runtime environment.

For more information about failure recovery, see "Summary of Process Interactions" in Chapter 3, "Netscape Application Server Architecture." See also the *Administration Guide* or the *Deployment Guide*.

For more information about load balancing, see "Load Balancing System" in Chapter 3, "Netscape Application Server Architecture." See also the *Administration Guide* or the *Deployment Guide*.

Session and State Management

Netscape Application Server supports state and session management capabilities required for web-based applications. NAS provides a number of classes and interfaces that application developers can use to maintain state and user session information.

State and session information is stored on each server in a distributed environment. For example, an application can display a login screen, prompt users to enter their user name and password, then save this information in a session object. Thereafter, the application uses this same information to log in to multiple databases without prompting the user to type it in again.

Similarly, in an online shopping application, a session object can store a list of products selected for purchase (such as quantity, price, and so on) and persistent variables (such as running order totals).

State and session management is especially important for applications that have complex, multi-step operations. In an environment where application logic is partitioned across different servers, system administrators can use the Netscape Application Server Administrator to optionally designate a single server to act as the central repository for all state information.

For more information about session and state management, see the *Programmer's Guide*.

Security

Netscape Directory Server provides NAS applications with data security by using access control lists, Secure Sockets Layer (SSL), and password policies.

In addition, NAS provides secure web server communication and supports SSL, HTTPS, and HTTP challenge-response authentication. To bridge the security gap between browsers and data sources, NAS supports user authentication, cookies, and database access controls for the secure handling of transactional operations. Event logging and tracking enables detection of, and protection against, unauthorized access.

User Authentication

NAS Administrator and Netscape Directory Server provide facilities to enable user authentication to ensure that only authorized users can access applications, databases, and directories.

Server administrators can use the security tool of the Administrator to create users, groups, and roles to manage access to specified resources.

Access Controls to Data Sources

Netscape Application Server works within the framework of existing access controls for relational database management systems. A user or application must log into the database before gaining access to the data.

Developers can write applications so that users enter login information only once and the application saves the information in a session object. Thereafter, the application uses the initial login information to log into different databases, as needed, in the background without requiring additional user input.

Netscape Application Server shields back-end data by acting as a secure gatekeeper between the web server and the relational database system.

Management Capabilities

Netscape Application Server eases application management by providing integrated management facilities. These facilities include the following:

- Netscape Application Server Administrator

- Netscape Directory Server

- Support for Third-Party Management Tools

Netscape Application Server Administrator

Netscape Application Server (NAS) Administrator is a Java application with a graphical user interface. NAS Administrator enables the following capabilities:

- remote management of multiple servers and distributed applications.

- dynamic deployment and scaling of applications.

- performance tuning and optimization of the server environment.

Management and tuning involves tasks such as adjusting database connection threads, adjusting load-balancing parameters, configuring web servers, and managing access control lists (ACLs).

Additional features of NAS Administrator include:

- Dynamic Application Management

- Event Logging and Failure Analysis

Dynamic Application Management

Netscape Application Server's architecture allows partitioned applications to run even if one or more servers fail. In a load-balanced server configuration, application logic can be replicated on multiple servers. If a server fails, the load balancing module dynamically directs requests to other available servers, thus preventing application-wide failure.

Because the NAS architecture promotes high availability of applications, server administrators can use NAS Administrator to perform a variety of tasks in real time, without interrupting an application's operation. These tasks include:

- monitoring, reconfiguring, or replacing servers.

- swapping out or updating application components.

Event Logging and Failure Analysis

Netscape Application Server provides facilities for logging requests from Web servers and logging system-level and application-level events on Netscape Application Servers. For deployed applications, system administrators can use contemporaneous logs to assist with failure analysis and to detect attempted security breaches.

Event logging occurs in multiple ways on the Netscape Application Server:

• Application developers can enable logging in their application logic to assist with failure analysis. For example, an application can log messages like "Transaction succeeded" or "Transaction failed" depending on conditions and events at run-time.

• System administrators can enable automatic event logging to record the messages generated by dynamically loadable modules (DLMs) and application execution.

• System administrators can enable HTTP request logging to record and monitor the requests received by a Web server. Administrators can specify brief, normal, or detailed logging. If logging is enabled, Netscape Application Server logs information about the HTTP requests to a specified target database. Administrators can then analyze the logs, generate custom reports, and so on. HTTP request logging requires NSAPI or ISAPI Web Connectors.

Netscape Directory Server

Netscape Directory Server, which is packaged with NAS, is Netscape's implementation of the Lightweight Directory Access Protocol (LDAP). NAS uses Netscape Directory Server not only to store NAS configuration data but also as a central repository for user and group information. A single Netscape Directory Server can support multiple instances of NAS—up to five clusters, in fact. This means that administrative data for all NAS installations can be centralized in one place.

The NAS Administrator acts as an LDAP client and can access information about users and groups. As a result of this integration with LDAP, NAS provides unified management of users, groups, and ACLs across the enterprise.

Support for Third-Party Management Tools

NAS provides the ability to be monitored and managed via SNMP agents. SNMP is a protocol used to exchange data about network activity.

NAS stores variables pertaining to network management in a tree-like hierarchy known as the server's management information base (MIB). Through this MIB, NAS exposes key management information to third-party tools that run SNMP. As a result, NAS can integrate with an enterprise's server management tools, thereby allowing other solutions for remote administration.

Netscape Application Server Architecture

This chapter describes the processes, systems, and services that make up the architecture of NAS.

The chapter contains the following sections:

- Server Processes

- System Components

Server Processes

The architecture of NAS includes four internal servers, which are often called engines or processes. They are responsible for all processing within Netscape Application Server. The following table summarizes the internal servers:

Internal Server	Process Name	Description
Executive Server	KXS	Provides most system services (some services are managed by the Administrative Server).
Administrative Server	KAS	Provides system services for NAS administration and failure recovery.
Java Server	KJS	Provides services to Java applications.
C++ Server	KCS	Provides services to C++ applications.

Summary of Process Interactions

The following figure shows how the four NAS processes interact to handle requests from clients:

Web browsers   Web servers   Netscape Application Servers
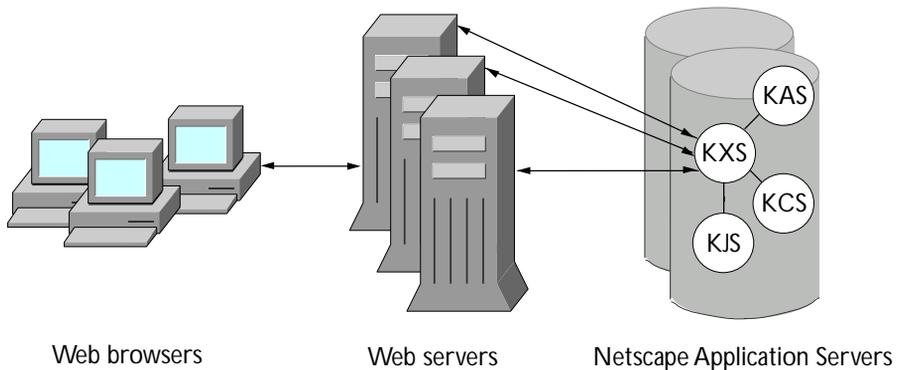
When a web server forwards requests to NAS, the requests are first received by the Executive Server process (KXS). The KXS process forwards the request either to a Java Server process (KJS) or to a C++ Server process (KCS). A KJS process runs Java programming logic, whereas a KCS process runs C++ programming logic.

Each KJS and KCS process maintains a specified number of threads and runs the programming logic to completion on those threads. The results are returned to the web server and sent on to the client browser.

The NAS technique of processing application requests is the key to reducing the load on a web server, thereby providing faster response time. A server administrator can configure the NAS environment for best performance by

- adding any number of NAS machines.

- specifying any number of KCS and KJS processes.

- maintaining any number of threads on each process.

In addition to providing high performance, the internal processes make it possible for NAS to remain available 24 hours a day, 7 days a week. If a KCS or KJS process goes down, the KXS process restarts it. And if the KXS process itself fails, then it is restarted by an additional process—the Administrative Server process (KAS). Additional monitoring in NAS makes sure that the KAS process is always running.

If all NAS processes go down, then other NAS machines in the cluster will take over. (This assumes a multiserver environment.) In addition, NAS can send notifications by email and FAX to alert the system administrator.

The next several sections describe the internal servers in more detail.

The Executive Server

The Executive Server is the main engine in the Netscape Application Server. The Executive Server is responsible for hosting many of the system-level services as they are needed by NAS.

The Executive Server process (KXS) also distributes application requests to the appropriate application process, either the Java Server or C++ Server process.

For example, here is what happens when an application request comes into Netscape Application Server:

1. The Executive Server invokes the request manager, a system-level service.

2. The request manager assigns a thread to the request and forwards the request to the appropriate application process, which is either the Java Server or the C++ Server.

3. When the request manager is finished, it is dismissed from the Executive Server, providing a dynamic-usage model for increased server performance.

The Administrative Server

The Administrative Server enables administration of one or more Netscape Application Servers. It registers with the appropriate server or servers all changes made to the system and application settings using Netscape Application Server Administrator (NASA), the GUI administration tool.

The Administrative Server also hosts the failure-recovery service that restarts the other server processes if they become unavailable. This failure-recovery service provides a high degree of fault tolerance for Netscape Application Server.

The Java Server and C++ Server

The Java Server and C++ Server processes are the application servers. Business logic components written in Java are hosted in the Java Server, whereas components written in C++ are hosted in the C++ Server. Business logic components are the core of the application, holding the compiled code instructions written by the developer.

The Java Server and C++ Server processes (KJS and KCS) also host the application-level services. These services are dynamically loaded into the appropriate process as needed by an application component. For example, when an EJB requires access to a database, the Java Server loads in the data access engine, uses its services, and then dismisses it. The database connection provided by the data access engine is cached in the Java Server active memory. If another request enters the system and accesses the same database, the cached connection is used. In this way, NAS reduces the need to invoke the data access engine, thereby increasing the performance of request processing.

System Components

This section describes the main internal systems that make up the NAS architecture. These systems are shown in the following figure:

Netscape Application Server
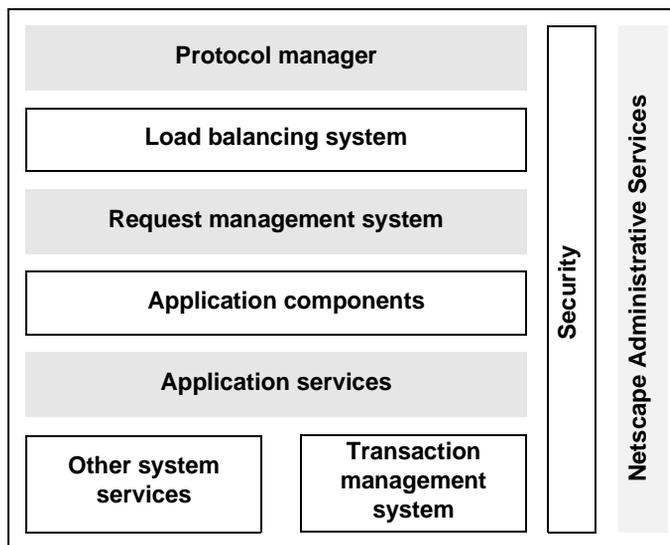
This section describes the following topics:

- Protocol Manager
- Load Balancing System
- Request Management System
- Application Components
- Application Services
- System Services
- Transaction Management System
- Security
- Administrative Services

Protocol Manager

Communication between a web server and Netscape Application Server occurs through NSAPI, ISAPI, and optimized CGI. Optimization and superior performance are achieved through listeners, web connectors, the streaming service, and the Protocol Manager, as described here:

- Web connectors and listeners manage the passing of requests from the web server to the Netscape Application Server. Listeners distribute and handle requests from the web connectors. New listeners can be added with the HTTP handler.

- The HTTP streaming service handles data streams from the Netscape Application Server to the web server and to the web browser.

- The NAS Protocol Manager manages and provides services for all active, loaded listeners. The Protocol Manager supports HTTP, HTTPS (HTTP over SSL), and IIOP.

The following figure shows how the Protocol Manager enables communication between NAS and a client (whether web-based or not):
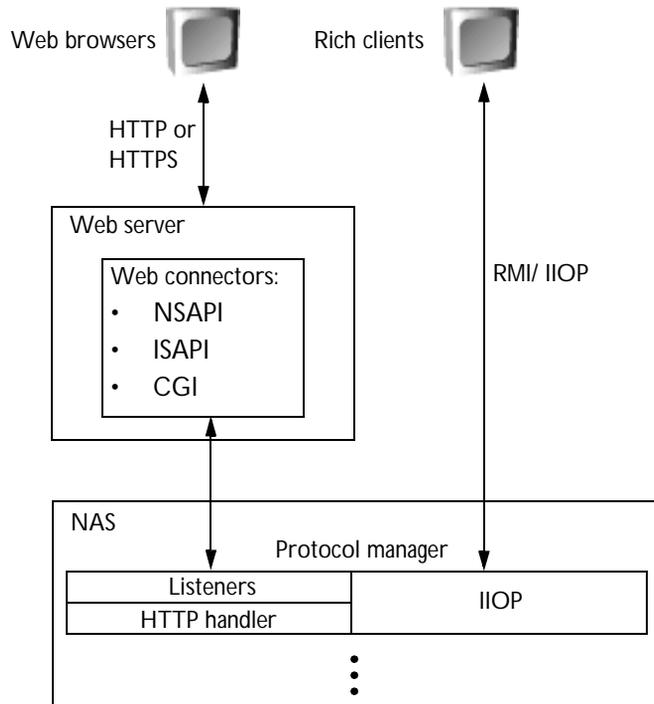
When a request comes in from a web browser, the request is passed to the web server via the HTTP or HTTPS protocol. The request is processed by the appropriate web connector. Web connectors include the NSAPI web connector, ISAPI web connector, and optimized CGI web connector for Netscape, Microsoft, and CGI-compatible web servers, respectively.

The request is then forwarded to the corresponding listener on NAS. Listeners on NAS include NSAPI listeners, ISAPI listeners, and optimized CGI listeners that communicate with Netscape, Microsoft, and CGI-compatible web connectors, respectively.

Load Balancing System

In an environment with multiple NAS installations, incoming requests first pass through the Load Balancing System. The Load Balancing System directs the request to the server best suited to process it. The Load Balancing System includes a Load Monitor and a Load Balancer.

The Load Monitor takes a snapshot of server load based on administrator-configured interval settings. The Load Monitor then tabulates resource availability and system performance.

The Load Balancer uses the load and performance statistics retrieved by the Load Monitor to determine the server with the most available resources to handle an incoming request. Each instance of NAS has its own load balancing module which makes routing decisions based on load balancing parameters. Administrators can tune these parameters using NAS Administrator.

After a request is routed to the appropriate NAS machine, the request is passed to the Request Management System.

Request Management System

Incoming requests are handled by the Request Management System. NAS is multi-threaded, and the Request Management System assigns threads from a dynamic thread pool to process the requests. The Request Management System enables the simultaneous processing of a high volume of requests. System

administrators can configure thread pool parameters for optimal request processing. The Request Management System includes the following subsystems:

- Thread Manager

  The Thread Manager provides a dynamic pool of threads. From this pool, a thread is assigned to process the request.

- Queue Manager

  The Queue Manager gets involved when requests must be queued until a thread becomes available.

  The Queue Manager manages the list of pending requests and descriptive information. This information includes things such as the unique request ID and a request's current processing status, such as waiting, in process, finished, and so on.

- Request Logging

  Request Logging, if enabled by the system administrator, keeps an information log of web server requests in a back-end database or in log files.

Application Components

Each incoming request identifies one or more application components. These components, in turn, are identified by their globally unique identifier, or GUID. GUIDs are checked against NAS's Global Directory Service (GDS) and Netscape Directory Server, an LDAP server.

GDS determines the language of the application component and then loads the appropriate language-specific handler for the request. For example, on Windows NT, a `.class` file is loaded for Java application logic, and a `.dll` is loaded for C++ application logic. On UNIX, a `.class` file is loaded for Java application logic, and a shared library (a `.so` file) is loaded for C++ application logic.

If necessary, Netscape Directory Server authenticates the request by checking against known access control lists (ACLs).

The appropriate application component is then executed to process the request. For example, the request may invoke a servlet, which in turn may call one or more EJBs.

Typically, request processing involves calling functions from among the application services, transaction management system, or other system services. For example, the application may make use of state and session management or may connect to a database to update a user's account.

For more information about application components, see "Application Model" in Chapter 2, "Features." In addition, see the *Programmer's Guide*.

Application Services

Application services enable management of application functions, such as user sessions, application states, cookies, email notification, result caching, and so on. These services are invoked by application components using API calls. Application services are loaded into either a KJS process, a KCS process, or both. The following sections summarize the services available to Java and C++ applications:

Services Hosted by Either KJS or KCS

The following services are available to applications written in Java or C++:

Application Service	Description
State and session management	Manages user session information, such as user login, page navigation information, and "shopping cart" selections. Manages persistent state information. Distributed NAS machines can use a state workspace to share information.
Cookie management	Generates HTTP cookies for cookie-aware web browsers. For non-cookie aware browsers, emulated cookies are embedded in URLs or hidden fields.
Data access management	Provides and manages access to databases.
Transaction management	Manages database transactions, providing commit and rollback support for those transactions. See "Transaction Management System" for more information.

Application Service	Description
Template management	Manages the merging of result-set data with templates before data is returned to the user.
Database connection caching	Caches database connections so that future access for the same database is provided immediately.
Result caching	Caches result-set data so future requests can be processed more efficiently. If the request has been stored in the result cache, the previously computed result is returned immediately. Otherwise, the application logic is executed and the result is processed. System administrators can configure result cache settings such as the number of cache slots, time-outs, and cache cleaning interval.
Application events	Based on time criteria or other event criteria, allows applications to send and receive emails, to invoke an AppLogic, or (for Java applications) to invoke a servlet. This is useful for administration.
HTML streaming	Provides streaming of data back to HTML clients so as to return data more efficiently.
Extensions	Allow enterprise applications to integrate with applications deployed on NAS. Extensions are persistent modules that are dynamically loaded into NAS and are accessed by multiple AppLogics or EJBs over the life of the extension. Although extensions act as application services, extensions can also be considered as application components.

Services Hosted by KJS Only

The following services are available only to applications written in Java:

Application Service	Description
JSP compiler	Interprets JSP tags, the HTML-like tags that determine the layout of pages sent to a web browser. The compiler supports Version 0.92 of the JavaServer Pages specification.
Servlet container	Contains and manages servlets through their life cycle by providing network services over which requests and responses are set, by decoding MIME-based requests, and by formatting MIME-based responses. Supports HTTP and HTTPS.

Application Service	Description
EJB container	Provides a home for EJBs and manages the beans it contains. Management involves registering beans, providing a remote interface for them, creating and destroying instances, checking security, managing their active state, and coordinating distributed transactions. The EJB container can also manage all persistent data within the bean and includes a full global transaction manager.
Distributed transactions	Supports transactions involving multiple databases (of different types or in different locations). A distributed transaction is invoked from an EJB and uses the built-in transaction processing manager of NAS.
LDAP support	Eases management and security by providing a central repository for information about users, groups, and access control lists. LDAP clients that ship with NAS include NAS Administrator and Netscape Directory Server.

System Services

System services increase the efficiency with which application requests are processed. These services are not directly used by applications, but rather provide additional application support outside the scope of application logic. There is no API access to system-level services. A description of these services is provided in the following table. (Note that some of the following services are described elsewhere in this chapter.)

System Service	Description
Protocol management	Manages communications with clients by supporting the various protocols used by the Netscape Application Server.
Request management	Manages requests as they arrive at the server, routing them to the proper processes (either the Java Server or the C++ Server) and managing request thread allocations.
Global Directory Service	Determines the programming language of an application component so it can be loaded into the appropriate server process, KJS or KCS.

System Service	Description
JNDI	The Java Naming and Directory Interface (JNDI) is a standard extension to the Java platform. The JNDI API provides Java applications with a unified interface to multiple naming and directory services in the enterprise.
Event logging	Maintains a log of application logic execution. Application developers can enable logging in their application logic to assist with debugging and tuning. In addition, system administrators can enable automatic event logging, which records the messages generated by dynamically loadable modules (DLMs) and application logic objects when processing user requests. Event logging can run in all processes.
Load balancing	Determines how application request loads are balanced among multiple servers.
Application request caching	Caches application requests so that future requests for the same application components by the same user are handled immediately.
Asynchronous processing	Provides processing of multiple requests at the same time when they arrive at varying times.
Failure recovery	Restarts the Executive Server, Java Server, or C++ Server processes if they ever become unavailable.
Distributed data synchronization	Supports failure recovery by synchronizing data. Data is synchronized not only across all KJS or KCS processes running in NAS, but also across all NAS installations within a cluster.
SNMP support	Provides access to NAS via SNMP agents, thereby allowing remote management from third-party administration tools.
Kernel services	Provide low-level services to all other services and subsystems. Examples of kernel services include language-binding engines and the lock manager.

Transaction Management System

The transaction management system supports access to back-end databases.

- C++ applications access databases using classes and interfaces provided in the NAS API, also called the Foundation Class Library.

- Java applications access databases using the standard JDBC API.

Local versus Global Transactions

For Java applications, NAS supports both local transactions and global (also called distributed) transactions.

Global transactions span multiple databases (optionally of different types). Global transactions occur using a two-phase commit from Encina, a transaction manager built into NAS. Local transactions involve access to a single database. They provide better application performance because they are less complex.

Global transactions can only be begun declaratively through EJBs. By contrast, local transactions can only be begun programmatically, from either servlets, JSPs, or EJBs.

Architectural Details

The following figure shows the architectural details of the transaction management system:
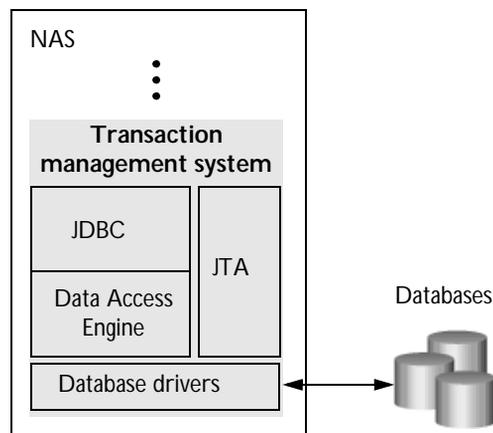
Both the JDBC and NAS APIs rely on the Data Access Engine to interact with database drivers. NAS provides native support for the following database drivers: Oracle, DB2, Informix, Sybase, and (on Windows NT only) SQLServer.

The transaction management system also includes the Java Transaction API (JTA). JTA is used for connections to a single database. JTA specifies local Java interfaces between the transaction manager the other transaction elements (which include NAS and the transactional application). JTA provides a Java mapping of the industry-standard X/Open XA protocol, which is used by the Encina transaction manager.

For more information about accessing databases, see the *Programmer's Guide*.

Security

In NAS, security is supported across all subsystems through the use of LDAP and other secure protocols. For more information, see "Security" in Chapter 2, "Features." See also the *Administration Guide* or the *Programmer's Guide*.

Administrative Services

Administrative services run in KAS, the Administrative Server process. KAS enables remote administration of servers and applications. KAS also supports other services, such as application partitioning, event logging, request monitoring, and dynamic configuration of key server settings.

Clients that access administrative services include NAS Administrator, Netscape Directory Server, and third-party SNMP agents. For more information, see "Management Capabilities" in Chapter 2, "Features." In addition, see the *Administration Guide*.

Index

A

Administrative Server (KAS) 33, 34
APIs 11
application events 40
application model 16
application partitioning 19
AppLogics 17

C

C++ Server (KCS) 32, 34
caching
　database connections 22, 40
　results 22, 40
client tier 10
cookie management 39

D

data streaming 23
data tier 10
deployment 13
distributed transactions 41

E

EJB container 41
Enterprise JavaBeans 17
event logging 28, 42
Executive Server (KXS) 32, 33
extensions 18, 40

F

failure recovery service 34, 42

G

Global Directory Service (GDS) 41

J

Java Naming and Directory Interface (JNDI) 42
Java Server (KJS) 32, 34
JavaServer Pages 17
JSP compiler 40

K

KXS, KAS, KJS, KCS 32

L

LDAP 41
listeners 36, 37
load balancing 20, 37, 42

M

multi-threading support 23
multitiered environment 9

N

Netscape Application Server Administrator 27
Netscape Application Server Deployment
　　　　Manager 13
Netscape Directory Server 28
Netscape Extension Builder 18

P

protocol manager 35

Q

query files 17

R

request management 37

S

sample applications 12
security 25
server processes 32
server tier 10
services
  application services 39
  system services 41
servlet container 40
servlets 17
session management 25, 39
state management 25, 39
streaming 23

T

templates 17
transaction management 43

W

web connectors 36, 37