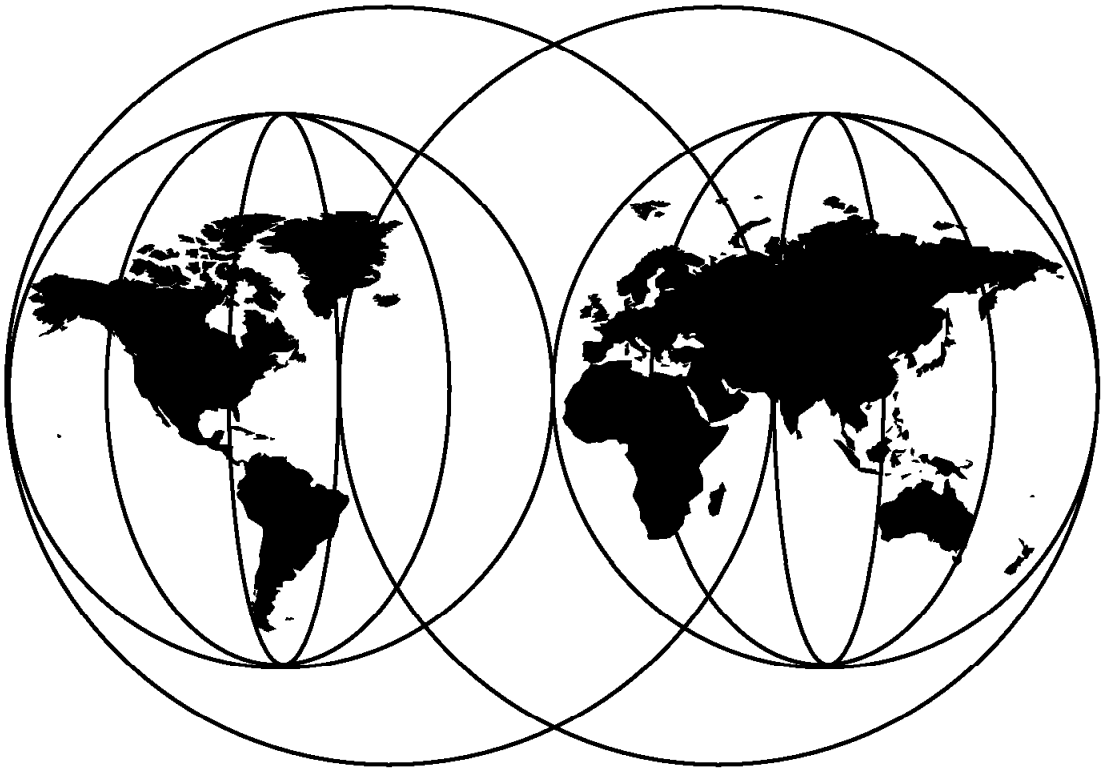
AS/400 e-commerce: Internet Connection Servers

Mick Lugton, Heikki Arhippainen, Saori Fujioka, Lee Hargreaves, Jens-Christian Vogt



International Technical Support Organization

http://www.redbooks.ibm.com

SG24-2150-00

International Technical Support Organization

SG24-2150-00

AS/400 e-commerce: Internet Connection Servers

April 1998

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix A, “Special Notices” on page 233.

First Edition (April 1998)

This edition applies to Version 4 Release 1 Modification Level 0 of TCP/IP Connectivity Utilities for AS/400, Program Number 5769-TC1; Internet Connection Secure Server (US and Canada), Program Number 5769-NC1; and Internet Connection Secure Server (International), Program Number 5769-NCE, for use with the OS/400.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. JLU Building 107-2
3605 Highway 52N
Rochester, Minnesota 55901-7829

Contents

Preface

This redbook studies the new (OS/400 V4R1) Internet Connection Servers. Internet Connection Server for AS/400 is the ″free″ one and replaces the V3R2/V3R7 HTTP server. Internet Connection Secure Server for AS/400 provides a secure server. Both servers provide support for multiple IP addresses, multiple server instances, configuration through a Web browser and access control. Internet Connection Secure Server provides support for secure HTTP transactions through the SSL (Secure Sockets Layer) protocol. This redbook covers the use and configuration of both servers. For the secure server, the redbook includes certificate management, getting a certificate from a certificate authority, creating a self-signed certificate, and so on.

The intended audience for this redbook is AS/400 Technical Specialists and Network Computing Specialists working with, or planning to work with, the AS/400′s Internet Connection Server support.

The book is unique in providing information both on the level of support provided and information on how to install/configure Internet Connection Server for AS/400.

The Team That Wrote This Redbook

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

Mick Lugton is a Systems Engineer at the Systems Management and Networking ITSO Center, Raleigh. He has nine years of experience in AS/400 communications and networking. He writes extensively and teaches IBM classes worldwide on all areas of AS/400 networking and communications. Before joining the ITSO three years ago, Mick worked in the AS/400 Business Unit, IBM U.K. as a networking specialist.

Heikki Arhippainen is a Services Specialist in Finland. He has 15 years of experience in the S/38 and AS/400 field. He has worked at IBM for seven years. He holds a degree in computer science from the University of Helsinki. His areas of expertise include communications, performance, and communications programming. He has written and lectured extensively on the Internet Connection Server for AS/400.

Saori Fujioka is an I/T Engineer in Japan. She has three years of experience in the AS/400 field. Her areas of expertise include TCP/IP. She has written and lectured extensively on the Internet Connection Server for AS/400.

Lee Hargreaves is a Technical Support Specialist in the U.K. He has eight years of experience in the communications and networking field. He has worked at IBM for nine years. His areas of expertise include TCP/IP and APPC communications. He has written extensively on SNMP, TCP/IP, WWW, and the Internet.

Jens-Christian Vogt is a Senior System Consultant with Norwegian Insurance Computer Environment in Norway. He has nine years of experience in AS/400 operations, communication, and security. His areas of expertise include database design, transaction security, and system operations.

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

Frank Paxhia
AS/400 Web Server Development, IBM Endicott

Jeff Remfert
AS/400 Cryptographic Services, IBM Rochester

Terry Hennessy
AS/400 Cryptographic Services, IBM Rochester

Comments Welcome

Your comments are important to us!

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

- Fax the evaluation form found in "ITSO Redbook Evaluation" on page 243 to the fax number shown on the form.
- Use the electronic evaluation form found on the Redbooks Web sites:

  For Internet users `http://www.redbooks.ibm.com/`
  For IBM Intranet users `http://w3.itso.ibm.com/`

- Send us a note at the following address:

  `redbook@us.ibm.com`

Chapter 1. Introduction

This redbook is intended to help you configure an AS/400 system as a Web server using the Internet Connection Server/400 (ICS/400) and Internet Connection Secure Server/400 (ICSS/400) products.

In this chapter, we introduce the concepts of the Internet, Web server, and the technologies associated with a Web server.

1.1 The Internet

The Internet is *the* network of networks. It is an interconnection of many smaller networks, logically linked using the Internet Protocol (IP) addressing scheme, to form a single larger entity, the Internet.

The Internet supports communications using applications and protocols from the Transmission Control Protocol/Internet Protocol (TCP/IP) suite and other compatible IP protocols. To put it simply, the Internet is a TCP/IP network and supports TCP/IP applications.

Although the origins of the Internet can be traced back to the late 1960s, it was in 1982 that the term Internet was born. Popular TCP/IP applications such as File Transfer Protocol (FTP) and electronic mail (e-mail) were, and still are, responsible for a phenomenal growth in Internet usage. Since 1992, the World Wide Web (WWW) has been responsible for the exponential growth in Internet usage and interest.

1.1.1 Intranets

An intranet is considered to be a private, or corporate, IP based network. That is, a TCP/IP network not connected to the Internet or logically separated from the Internet by means of a Firewall. Because an intranet is based on the same TCP/IP protocol suite as the Internet, applications used over the Internet can be used over an intranet, and vice-versa.

1.2 The World Wide Web (WWW)

The World Wide Web is a client-server based Internet application. It grew out of research by CERN, the European Laboratory for Particle Physics, during the 1980s. Originally developed for use within CERN, the WWW specifications were made public in 1992. In 1993, the National Center for Supercomputing Applications (NCSA) developed a graphical client interface to the WWW, what we now call a **Browser**, called *Mosaic*. At the end of 1993 there were only an estimated 130 WWW Servers on the Internet but, in 1994 the first commercial Web sites started appearing and by year-end, there were around 10 000 Web sites on the Internet. Current estimates indicate more than a 1 000 000 Web sites are in existence on the Internet and that the figure is doubling every six months.

In its simplest form a Web server serves documents (see Figure 1).

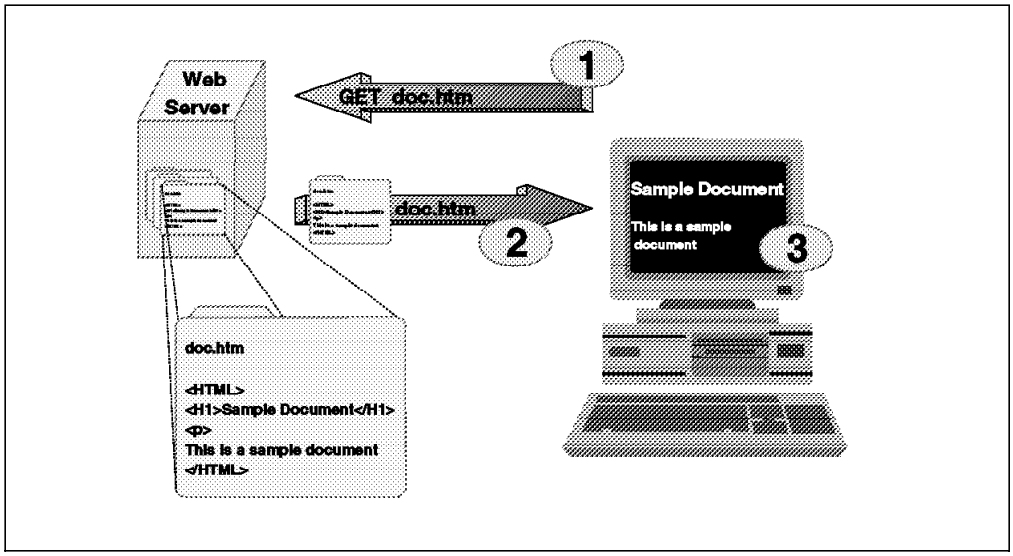


Figure 1. Simple Web Serving

1. A Web browser requests a document.
2. The Web server sends the requested document.
3. The Web browser interprets the document and displays it.

Hypertext Transfer Protocol (HTTP) is the underlying TCP/IP protocol used to retrieve the document.

WWW documents are written in a script language called Hypertext Markup Language (HTML). The HTML code describes, to the Browser, the look of the displayed document.

We can make the following statements:

- The Internet is a large network used by TCP/IP applications.
- HTTP Web servers are TCP/IP applications that facilitate the transfer of HTML documents between a server and client.
- The WWW is a high level TCP/IP application. HTML and HTTP Web servers are examples of functions used by the WWW.

1.3 Web Server Usage

A Web server can be used to serve a simple, static, text document to a browser. At the other extreme, it can be fully incorporated into the core of a business serving multimedia product catalogs, handling online customer orders, distributing software, and many other functions that may, at present, be performed using hardcopy, the postal service, or other time consuming and costly methods. There are distinct phases a business may go through as it develops Web applications of increasing complexity.

1. Static Web pages. These are Web pages with information that can be updated but are otherwise static. With static pages, all visitors to the Web site see the same content. An example of a static Web page is a company′s home page containing marketing and company contact information.

2. Dynamic Web pages. These are Web pages that can be customized to viewers who register at a site and specify their interests. This allows you to tailor your advertising to particular customer profiles. Dynamic Web pages can also be used to provide access and update capability to databases. For example, employees can use dynamic Web pages to access certain fields of their personnel records and update them.
3. Enhanced collaboration. In this phase, you have more direct interaction with the customer through discussion groups or forums that provide product support answers to customer questions. This phase may also incorporate advanced Web technologies such as chat, news groups, or audio support.
4. Full business integration. In this phase, Web-based applications are fully integrated into the core business. E-business or e-commerce are examples of this. From these sites, customers select and order products and services over the Web. They can also query order status, provide payment information, and select delivery options.

Each phase of Web presence builds on the earlier phase. In general, phases increase in complexity and difficulty. The rate at which the Web is maturing indicates that most customers reach the advanced phases within 15 months of establishing a Web presence.

There is one more aspect to the ″evolution of going online″. Usually companies do this first in an intranet environment, then and only when they feel comfortable, do they externalize their presence.

1.4 Web Server Functions

To implement a successful Web server, there are certain areas that should be considered.

Configuration File

Every Web server must have a configuration file of some description to tell the server how to behave. The CERN model configuration file is commonly used. A configuration file contains statements, or *directives*, which, when read, allow the server to determine its behavior. The file may contain directives pertinent to areas such as the name of the Web server, access control, or the port number at which the server should receive requests.

Access Control

Access control is normally implemented within the configuration file. It simply informs the server, by means of specific directives, which documents are permitted to be served to particular clients.

Access control on the AS/400 system is also achieved through normal OS/400 object control. That is, you can designate which user profiles have access to what objects. With Internet Connection Server for AS/400, we can use both access control through a configuration file and honor OS/400 object security.

Logging Logging can be used to keep track of who is accessing the Web site. Logs can be analyzed to find out a variety of statistics such as how many times a particular page is accessed.

Security As more and more business is conducted over the Internet, it is becoming increasingly important to ensure privacy while conducting a transaction. For example, it may be undesirable to send credit card details to a Web server over the Web in the *open*. Encryption provides a method of ensuring that the details of a message or transaction are visible only to the client and server. Authentication provides a method of ensuring a server or client's identity before, for example, sending a credit card number to a server. One method of achieving privacy through encryption and authentication is by using the Hypertext Transfer Protocol Secure (HTTPS) and the Secure Sockets Layer (SSL) protocols.

1.5 AS/400 System and WWW

By utilizing the native TCP/IP support of the AS/400 system and the products *Internet Connection Server for AS/400* and *Internet Connection Secure Server for AS/400*, it is possible for the AS/400 system to become a fully-functional Web server on the Internet or at the center of a corporate intranet.

The goal of this redbook is to help you implement the server functions such that you can fully exploit the Internet or an intranet as efficiently and as effectively as possible. The book, however, does not cover all aspects of implementing a server function. An important subject that is not covered in detail in this book is network security. For further reading on network security, see *AS/400 Internet Security : Protecting your AS/400 from HARM on the Internet*, SG24-4929, and *AS/400 and the Internet*, G325-6321.

Chapter 2. Internet Connection Server Family

Internet Connection Server for AS/400 and Internet Connection Secure Server for AS/400 are members of the IBM Internet Connection Server family. Internet Connection Server family members are members of the Internet Connection product line that provides clients, servers, security, network services, and consulting. The family consists of Web servers that are available for several different IBM and non-IBM platforms. In this chapter, we briefly look at the IBM Internet Connection Server family members.

2.1 Internet Connection Secure Server

The IBM Internet Connection Secure Servers provide the capability to create a secure WWW presence on the Internet. The Internet Connection Server family provides consistent APIs, configuration, and administration across the spectrum of server platforms. Version 4.2.1 of the Internet Connection Secure Server is currently available for:

- AIX
- HP-UX
- OS/2 Warp
- OS/390 (Version 2.2)
- Sun Solaris
- Windows NT
- Windows 95 (beta version available)

Note: Internet Connection Secure Server Version 2.1 is a part of OS/390 Release 2 and Internet Connection Secure Server Version 2.2 is a part of OS/390 Release 3. ICSS V2.2 for OS/390 uses the same technology as ICSS 4.2.

Features of Internet Connection Secure Server up to Version 4.1 include:

- Can be accessed by any industry-standard browser
- Home Page repository
- HTTP support
- Proxy support and proxy caching
- CGI support
- Web-to-Database application development (Net.Data)
- SSL V2 and V3 support
- Easy-to-use configuration tool
- Interconnect Connection API (ICAPI)
- NSAPI support
- Cookies support
- Server-side includes
- Error-message customization
- Enhanced logging and reporting
- Multiple IP address support
- Provides a choice of nine languages for AIX, OS/2 Warp and Windows NT

Note: The Internet Connection Server for AS/400 and Internet Connection Secure Server for AS/400 include a subset of Internet Connection Secure Server V4.1 features. In addition, they also provide:

- Integration with OS/400
- Authentication through validation lists
- Multiple server support

- Browser based certificate tools
- Support for 51 languages

New features for Version 4.2.1 include:

- Customized response based on requesting browser
- Counter and date/time image support for Web pages
- Enhancements to logging and generating reports on server activity
- HTTP Version 1.1 compliance
- Expanded Common Gateway Interface (CGI) support
- Year 2000 support
- Java servlet support
- Performance improvements
- Web site content rating support, Platform for Internet Content Selection (PICS)
- Client authentication and other enhancements to SSL support
- Simple Network Management Protocol (SNMP) support
- SOCKS support and SSL tunneling
- Online access to server performance and status information
- Internet Connection Application Program Interface (ICAPI) support
- Proxy authentication
- In the OS/390 version:
  - OS/390 Workload Manager (WLM) support
  - OS/390 Cryptographic Hardware support
  - OS/390 dataset support
  - User-based statistics from Web Usage Mining

The Internet Connection Secure Servers V4.2 for AIX OS/2 Warp and Windows NT are available also in the following languages:

- French
- Spanish
- Italian
- German
- Brazilian/Portuguese
- Japanese
- Korean
- Simplified Chinese
- Traditional Chinese

For more information about IBM Internet Connection Servers, visit `http://www.ics.raleigh.ibm.com/icserver/`.

2.2 Domino Go Webserver

The latest versions of ICSS have been renamed to Domino Go Webserver. Domino Go Webserver is developed by IBM, but the brand name is owned by Lotus Development Corp. Domino Go Webserver should not be mixed up with Lotus Domino, which is a groupware and e-mail server. Domino Go Webserver is also packaged and marketed into Domino Go Webserver Pro. Included with Domino Go Webserver Pro are Lotus BeanMachine for Java, which allows the creation of exciting multimedia Java applets without programming; NetObjects Fusion, an advanced Web-site design and management tool featuring pixel-level control of Web pages and automatic maintenance of links; and Net.Data, for Web-to-database application development.

Domino Go Webserver is a part of the IBM Network Computing Framework that is a software roadmap for implementing e-business solutions. For more information about Network Computing Framework, see *Network Computing Framework (NCF)*, SG24-2119, or visit http://www.software.ibm.com/ebusiness/ncf/ncf-overview.html.

The current version of Domino Go Webserver is 4.6 and it is available for:

- AIX
- HP-UX
- OS/2 Warp
- OS/390
- Sun Solaris
- Windows NT
- Windows 95

Note: Domino Go Webserver 4.6.1 is orderable as a no charge feature of OS/390 Version 2 Release 4. IBM plans to integrate Domino Go Webserver 4.6.1 with OS/390 Version 2 Release 5.

Note: Domino Go Webserver will, in the future, also be available for the AS/400 system.

The features inherited from the Internet Connection Secure Server include:

- Home Page repository
- Full HTTP 1.1 compliance
- Repository for imbedded binary resources
- Proxy support and proxy caching
- CGI support
- Easy-to-use configuration tool
- NLS enablement
- Security
- Interconnect Connection API (ICAPI)
- Java servlet support
- Server-Side Includes
- Error message customization
- Enhanced logging and reporting
- Multiple IP address support
- Proxy authentication
- Local file caching
- Default code page support
- Online access to server performance and status information
- Web Usage Mining
- SSL V3 support
- Automatic browser detection
- CGI support for C, REXX, Perl, and Java
- Web-to-Database application development (Net.Data)
- PICS support
- Client authentication
- SNMP subagent
- SOCKS support
- SSL tunneling
- In the OS/390 version:
  - OS/390 Workload Manager (WLM) support
  - Use of the OS/390 System Authorization Facility (SAF)
  - S/390 Cryptographic Hardware support

- OS/390 Console Support
- OS/390 Dataset Support
- Inputs to OS/390 System Management Facility logs

New Features for Domino Go Webserver Version 4.6.1 include:

- Go Webserver API (GWAPI) support
- Authentication using a certificate
- Webserver Search Engine
- Java Beans support
- JIE Toolkit for Windows NT
- Performance Enhancements
- Proxy Enhancements
- Fast CGI

For more information about Lotus Domino Go Webserver, visit http://www.ics.raleigh.ibm.com/dominogowebserver/.

Chapter 3. AS/400 Internet Connection Server

Internet Connection Server for AS/400 (ICS for AS/400) provides Web server function for OS/400 V4R1. It is a follow-up product to the Hypertext Transfer Protocol (HTTP) server that shipped with TCP/IP Connectivity Utilities for AS/400 V3R2/V3R7. ICS for AS/400 is part of IBM TCP/IP Connectivity Utilities for AS/400 (5769-TC1). ICS for AS/400 will serve OS/400 V3R2 and V3R7 Web site content without change.

3.1 Introduction to ICS for AS/400

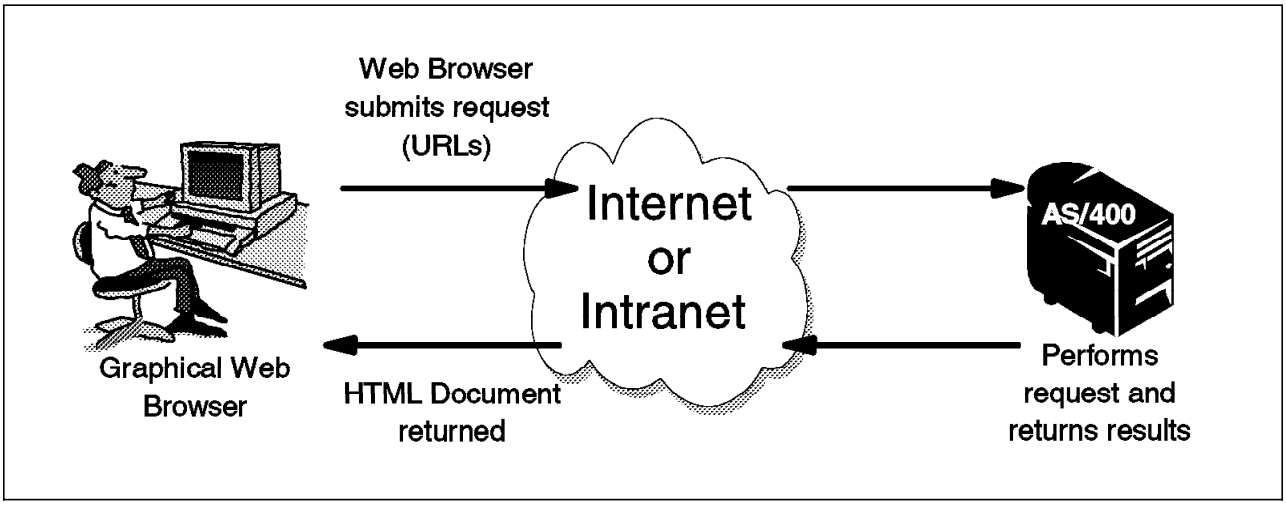


Figure 2. How the ICS for AS/400 Works

ICS for AS/400 enables serving multimedia objects such as HTML documents and their contents to Web browser clients. The objects served can reside in several file systems on the AS/400 system. ICS for AS/400 supports the following file systems:

"root" The / file system. This file system takes advantage of the stream file support and hierarchical directory structure of the integrated file system (IFS).

QOpenSys The open file system. This file system is compatible with UNIX-based open standards such as POSIX and XPG. In use, it is the same as the root file system, but the object names are case-sensitive.

QSYS.LIB The library file system. This file system supports the AS/400 library structure. It provides access to AS/400 database files and all of the other AS/400 object types that the library support manages.

QDLS The document library file system. This file system supports the folder structure and provides access to documents and folders.

QLANSrv The LAN Server/400 file system. This file system provides access to the same directories and files that are accessible through the LAN Server/400 licensed program.

QOpt The QOpt file system. This file system provides access to stream data stores on optical media.

QFileSvr.400 The QFileSvr.400 file system. This file system provides access to other file systems that reside on remote AS/400 systems.

For more information about the integrated file system, see the *Integrated File System Introduction*, SC41-5711.

Internet Connection Server for AS/400 includes support for the following:

- Access and error logging
- Restricting access
- Multiple server instances
- Multiple IP addresses
- Web applications

3.1.1 Access and Error Logging

To help determine whether or not an Internet message is reaching the intended audience, logs can be kept that show who is accessing a server and when. It is also possible to keep logs of internal server errors. What gets logged can be controlled by filtering out entries that have a host name or IP address that matches a particular pattern.

The logs are written either in Data Description Specification (DDS) or common format. When the logs are in DDS format, they are stored in the QUSRSYS library. When the logs are in common format, they are stored either in the QUSRSYS library or in one of the stream-based file systems on the AS/400 system. When in common format, the logs are written in a format that is used by most Web servers. Programs are commonly available that can be used to analyze logs in the common format. See Figure 3 on page 11 for an example of an access log analyzer output display.

The access log contains entries for each attempt to get to a page. For each access request a server receives, an entry is made in the access log showing:

- The date and time of access
- The client host name or IP address
- The server method and path to document accessed
- The remote login name of the user
- The user name (the name that has been authenticated)
- The HTTP status code returned to the client in response to the request
- The number of bytes transferred (the length of the document that was transferred)

The error log includes errors encountered by a server's clients such as timing out or denial of access. The log contains detailed information about the request that failed including time, error, address, and URL.

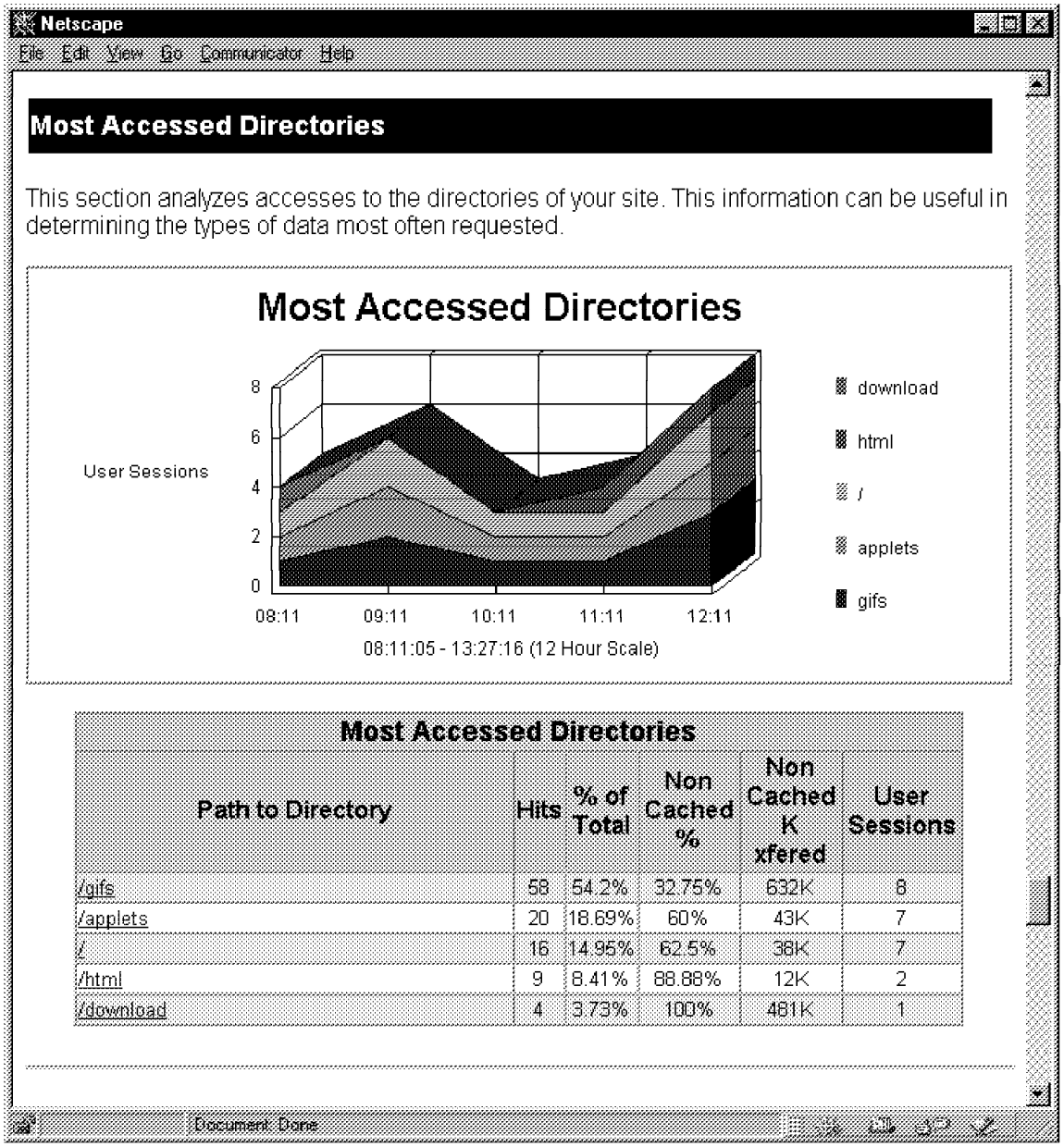


Figure 3. Example of Access Log Analyzer Window

The access log can be analyzed using several different readily available Web server statistics tools. Reports from these tools answer questions such as:

- Who is visiting the Web site?
- How long they are staying at the Web site?
- Which pages are the most popular?
- How many pages are accessed in each directory?
- How many users are accessing each directory?
- How many users visit the site daily?
- What paths visitors take when they browse the Web site?
- Which is the most active day of the week?
- Which is the most active hour?

3.1.2 Restricting Access

If required, access to a server can be restricted. Access can be restricted based on user name and password or the address of the requester.

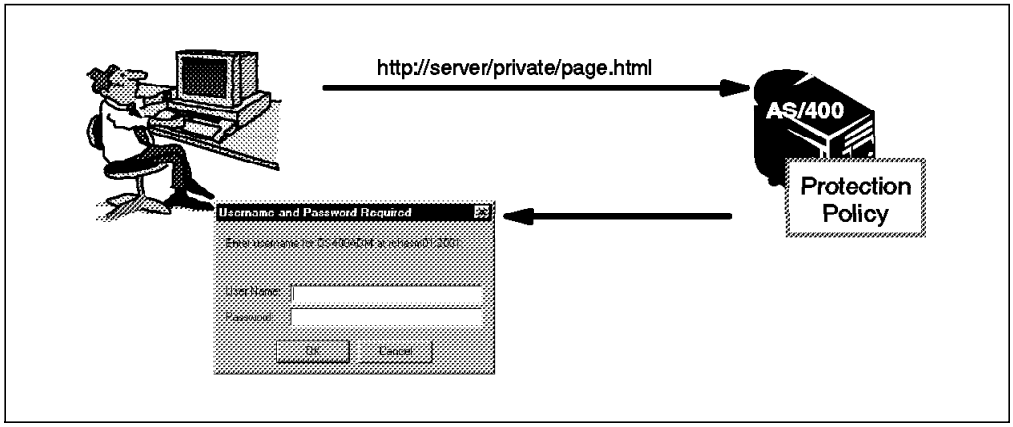


Figure 4. Protecting Server's Resources

Access authorization is controlled using the configuration file and possibly one or more other files, including:

- A protection setup that defines the protection rules being used
- A validation list in which user names and passwords are defined
- A group file in which groups of user names can be defined
- An Access Control List (ACL) file that allows the access to individual files or groups of files on a protected directory to be defined

3.1.3 Multiple Server Instances

ICS for AS/400 allows multiple instances of the server to be run. In effect, multiple servers with different configurations files can run on the same system. Multiple server instances allow different server content to be presented based on the server instance accessed.

Server instances can be configured to use unique IP addresses or unique port numbers. In Figure 5, there are three server instances, one for each network using the default port and one using a specified port number. SERVERA and SERVERB can be accessed by specifying only the host name in the URL, but SERVERC also requires a port number in the URL.

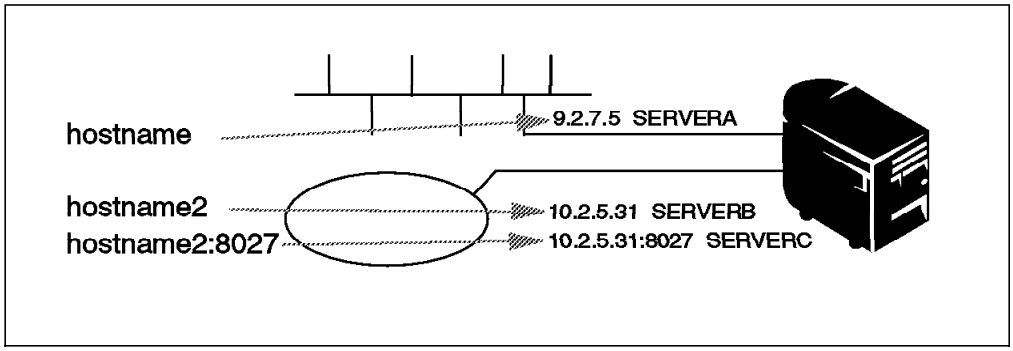


Figure 5. Multiple Server Instances

3.1.4 Multiple IP Addresses

ICS for AS/400 can be configured to serve different files based on the host name specified in the URL request. This can be valuable to Internet service providers who want to use one server to provide multiple Web sites for customers. Based on the host name specified in the URL request, the server allows the following options to be configured:

- Welcome pages to determine how the server responds to requests that do not contain a file name.
- Mapping rules that map requests to physical files and determine whether the server processes a request.
- Access control to activate different protection rules for requests depending on which address the request comes in on or which host name is specified in a URL.

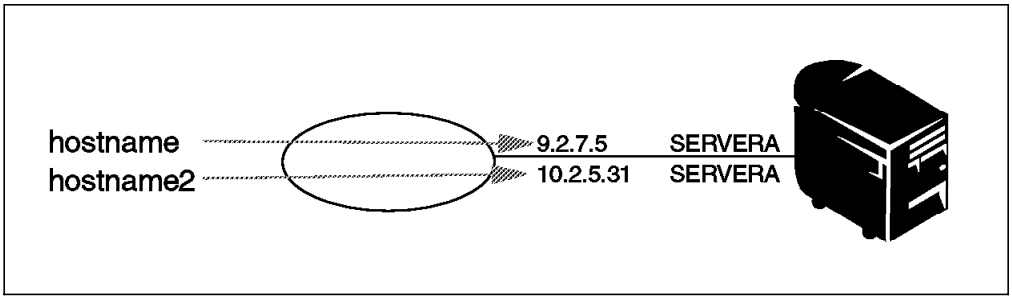


Figure 6. Running Server with Multiple IP Addresses

3.1.5 Web Applications

On the AS/400 system, it is possible to create HTML pages with dynamic content in several different ways.

The most prevalent programming interface is the Common Gateway Interface (CGI). The CGI interface is supported on the AS/400 system and can be written to using several integrated language environment (ILE) languages such as RPG, COBOL, and C.

A high-level macro language called Net.Data can also be used to develop dynamic Web pages. The highly functional Net.Data macros combine the simplicity of HTML and the functionality of CGI programs. Net.Data allows information stored in DB2 for AS/400 to be included by using standard SQL or by calling other programs or reading files. Net.Data is a follow-on to DB2 World Wide Web (DB2WWW) that was introduced to the AS/400 system in V3R2. The Net.Data in V4R1 is enhanced and improved over V3R7.

Figure 7. Web Application Example

For more information on Web application development for the AS/400 system, see *Unleashing AS/400 Applications on the Internet*, SG24-4935, and *ICS, ICSS Webmaster's Guide*, GC41-5434.

Chapter 4. AS/400 Internet Connection Secure Server

The Internet was not designed to handle commercial processes; the advent of e-shopping and e-business has, therefore, put forward a wide-spread demand for information protection and privacy. There is a need to not only protect the integrity of data but also to stop unauthorized persons from reading confidential information.

Any message passing through the Internet is open and the data within it can be read by anyone with a little knowledge on how the Internet protocols work (for instance, packet sniffing software is freely available on the Internet). There is no limit to the different software platforms that can read, copy, or alter any message sent.

This chapter provides a brief introduction to transaction level Internet security mechanisms, the Secure Sockets Layer protocol (SSL), and Hypertext Transfer Protocol Secure (HTTPS) as well as an overview of the components and services offered within V4R1 of the Internet Connection Secure Server for AS/400 (ICSS for AS/400) licensed programs 5769-NC1 (U.S. and Canada) and 5769-NCE (International).

4.1 Internet Transaction Security Concepts

When planning an e-business installation, there are several aspects of security that need addressing. Among these are:

- Server security, that is, the overall protection mechanisms required to secure the server applications and data. These include physical security, firewalls, operational routines, and more.
- Application security and integrity. The Internet does not bring any revolutionary new requirements into application security but existing requirements may have to be more strictly enforced and audited.
- Transaction security, or the protection of a single transaction and its contents.

Of these three, transaction security is the area covered in this chapter.

4.1.1 What is Transaction Security?

Do you send your medical records or pay your bills by postcard? Probably not, and for the same reasons few people use the Internet in its present form for commercial exchanges without additional security. But when is a commercial exchange secure? The answer to this and any other “when is something secure” question is - “it depends.” It depends on the business requirements, the company security policy, AS/400 configuration, and a multitude of other questions. The AS/400 architecture provides a strong set of security tools that have proven their value over a long period of time. But now we also have to learn about and implement network transaction security.

You can consider an information transaction to be secure if it has these characteristics:

- Confidentiality
- Integrity

- Accountability
- Authenticity

Let′s take a look at each of these and how to achieve them before we go into details about SSL, HTTPS, and the Internet Connection Secure Server for AS/400.

Important

If you are new to transaction security, some of the following concepts may be difficult to understand. However, it is important that you have a thorough understanding of the following concepts before you go on to Chapter 9, “Establishing a Secure Connection” on page 159.

4.1.1.1 Confidentiality

Confidentiality means that the contents of the messages remain private as they pass through the Internet. Without confidentiality, a computer broadcasts the message to the network, similar to shouting the information across a crowded room. *Encryption* is used to ensure confidentiality.

4.1.1.2 Integrity

Integrity means that messages are not altered while being transmitted. Any router along the way can insert or delete text or garble the message as it passes by. Without integrity, we have no guarantee that the message sent matches the message received. *Message digests* (or secure hash codes) ensure integrity.

Using the crowded room example, message integrity requires that the message be sent in a sealed envelope with a date and time stamp. In the electronic world of transaction security, we can achieve an even higher level of integrity control by using digital signatures which have mechanisms that verify the message was not tampered with during the transmission.

4.1.1.3 Accountability

Accountability means that both the sender and the receiver agree that the message exchange took place. Without accountability, the addressee can easily say the message never arrived. Accountability for the sender (or non-repudiation of origin) is handled by *digital signatures*. For the addressee, accountability is a bit more complex and involves automatic transmission of verification messages to the sender.

4.1.1.4 Authenticity

Authenticity means that you know who you are talking to and that you can trust that person. Without authenticity, we have no way to be sure that anyone is who they say they are. Authentication ensures identities (servers, clients, users). Using the crowded room example, message integrity requires that the message is encrypted using any “secret” language. The message must also be in a written form, signed by the sender, and the envelope sealed. If you are sure that the seal and the signature are authentic and not broken or tampered with, you may assume that the message is authentic. *Digital signatures* and *digital certificates* ensure authenticity.

All of the transaction security mechanisms can be executed both ways, that is, client-to-server and server-to-client.

4.2 Internet Transaction Security Methods

Now that we have covered the *concepts* of Internet transaction security (confidentiality, integrity, accountability, and authenticity), let's take a look at the *methods* required to implement these concepts into a live Internet (or intranet) environment. The methods are:

- Encryption
- Digital Signatures using Message digests
- Digital certificates

The Examples

On the following pages, there are illustrated examples of how the different transaction security mechanisms work. The examples deal with three persons:

Alice: She has two keys, a Private Key (APrivate) and a Public Key (APublic).

Bob: He also has two keys, a Private Key (BPrivate) and a Public Key (BPublic).

Charlie: He is an imposter trying to gain access to the information Alice and Bob are exchanging. Charlie has two keys as well, his Private Key (CPrivate) and his Public Key (CPublic).

Along the way, all three of them may get access to the public key of the other two.

4.2.1 Encryption

Encryption involves scrambling techniques so that the message being sent is unreadable to anyone without the correct unscrambling code (or decryption *key*). The original form of the message is known as *Plaintext* or *Cleartext*, and the scrambled message is known as *Ciphertext*.

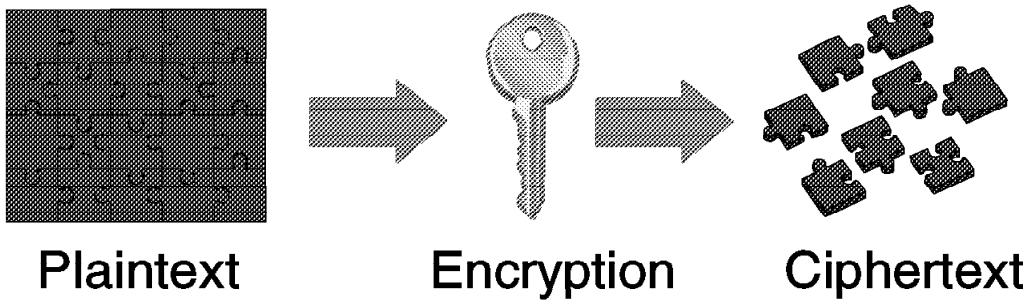


Figure 8. Plaintext, Encryption, and Ciphertext

If you are sure that no one in the previously mentioned crowded room understands Norwegian, you can encrypt your conversation by speaking Norwegian. This way, both participants use their knowledge of Norwegian as a shared secret or key, known as a *symmetric* key.

Symmetric keys pose problems in distributing and managing the keys in a safe manner. So we introduce the concept of *asymmetric* keys. Asymmetric keys involve what is known as a *key pair*.

A key pair consists of a *public key* and a *private key*; these two work together to achieve what the symmetric key does by itself:

- What you encrypt with a symmetric key, you can decrypt with the same symmetric key.
- What you encrypt with one part (private/public) of an asymmetric key pair, only the other part (public/private) of that asymmetric key pair can decrypt.

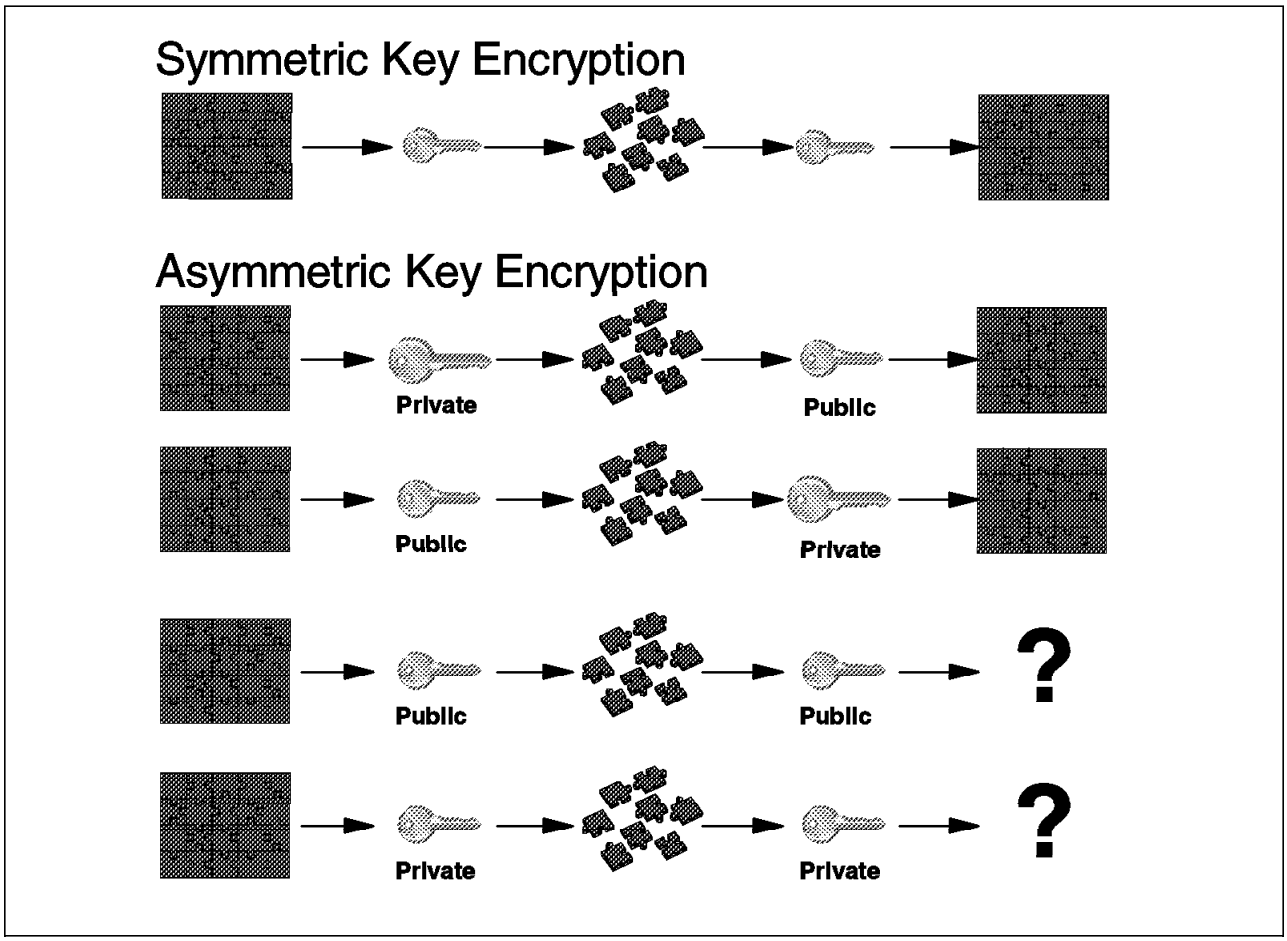


Figure 9. Symmetric and Asymmetric Key Encryption

The two parts of a key pair consists of two mathematically linked patterns, where each pattern or key is the inverse function of the other. The public key is made available to the general public and the private key is a kept secret. To repeat how these two elements of an asymmetric key pair works:

- A message encrypted by the public key can only be decrypted by the private key.
- A message encrypted by the private key can only be decrypted by the public key.

The Internet Connection Secure Server for AS/400 uses asymmetric keys to achieve initial confidentiality and switches to symmetric keys for data exchanges during the SSL handshake. The symmetric keys are disposable and are unique

to a data exchange session. For data exchanges, symmetric key encryption is more efficient than asymmetric key encryption.

There are a few tricky bits to the asymmetric key structure. One of the most obvious ones being the handing out of public keys. This is covered in 4.2.3, “Digital Certificates” on page 20.

4.2.2 Digital Signatures

Digital signatures use a “digital fingerprint” known as a *message digest* together with encryption to achieve both integrity and accountability. This message digest is based on the message sent and ensures that the message was not altered during the transmission.

The server generates the Message Digest using a message digest algorithm. The input to this algorithm is the message to be sent and the output is the 128-bit Message Digest.

The message digest is then signed; this is done by encrypting the message digest with your private key (this way it can only be decrypted using your public key). Figure 10 gives a simple example of a digital signature transmission.

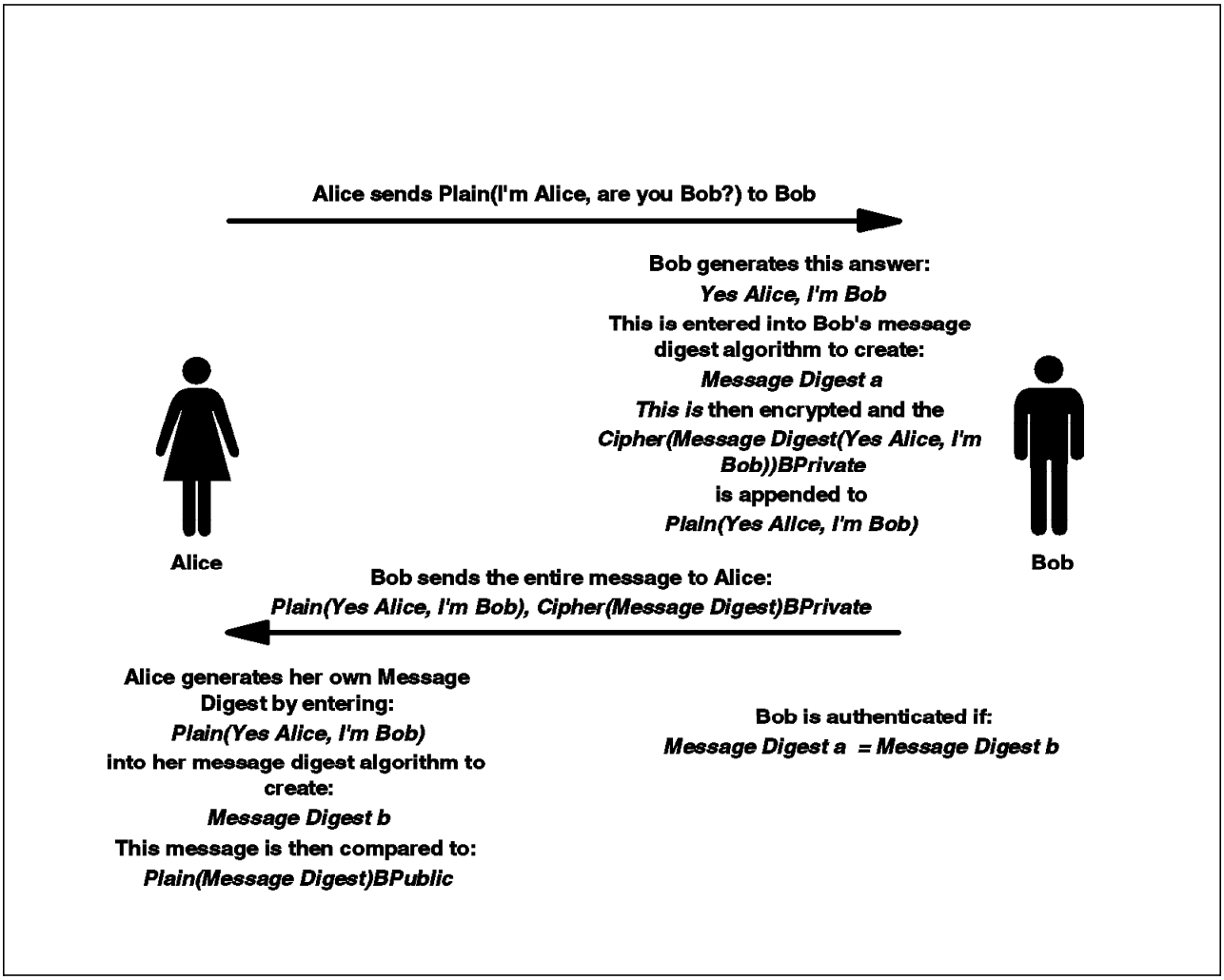


Figure 10. Bob Authenticates Himself with a Message Digest

Message String Syntax

Plain(text): Plaintext

Cipher(text)keyname: Plaintext encrypted with the appended “keyname” key.

Plain(text)keyname: Ciphertext decrypted with the appended “keyname” key.

Digest(text): The Message Digest of the “text.” The *Digest(text)* can also be inserted into encrypt/decrypt strings *(cipher(digest(text))keyname)*.

As an alternative to the preceding syntax, the message itself can also be encrypted. If the message is also encrypted, this is what happens: The plaintextmessage and the encrypted message digest is encrypted using the public key of the recipient. This is how the message looks:

Cipher((plaintextmessage), cipher(digest(plaintextmessage)BPrivate))APublic

By encrypting the message contents in this manner, we achieve this:

- The entire data string (message and message digest) is encrypted with the public key of the recipient, thus ensuring that it can only be decrypted with the recipients private key.
- The transmission also contains an encrypted message digest that can only be decrypted with the public key of the sender.
- When the original message digest is compared to the message digest computed by the recipient after decrypting the message and the two message digests are found to be identical, the recipient knows that the message was not altered in transmission.

4.2.3 Digital Certificates

Digital certificates handle authentication together with digital signatures. Authentication is the process used to verify identity. Digital signatures provide integrity and accountability, but how do we know that someone sending a message is who they say they are?

We look at the sender′s digital certificate. If you think of a digital signature as a credit card with your picture on it, then the digital certificate is the same as a credit card with a picture of the bank president with an arm around you. A merchant (or any third party) trusts you because you look the same as the picture on the credit card and the bank president trusts you also.

We base trust for the authenticity of the sender on whether we trust the third party that certified the sender. The third party or *Certification Authority (CA)* issues digital certificates. Digital certificates are typically made up of:

- The public key of the server being certified.
- The name and address of the server being certified, also known as the *Distinguished Name*.
- The digital signature of the CA.
- The issue date.
- The expiration date.

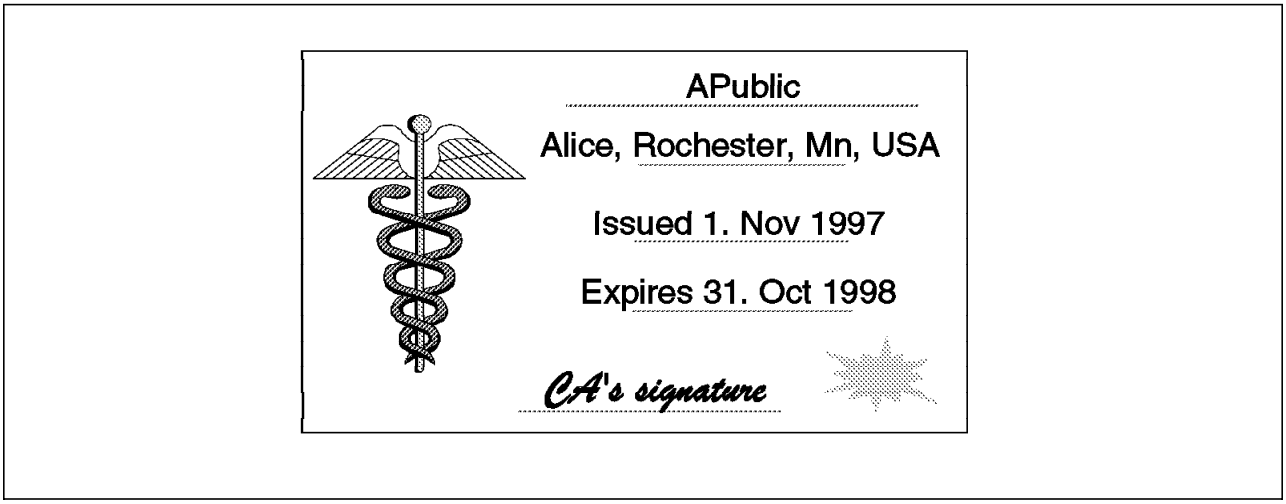


Figure 11. Alice's Certificate

A CA broadcasts its public key and distinguished name bundled together so that people add the CA as a *trusted root* key to their Web servers and browsers.

When you designate a Certification Authority trusted root key, you tell your server or browser that it can trust anyone with a certificate issued by that Certification Authority. A server or browser can have many trusted roots.

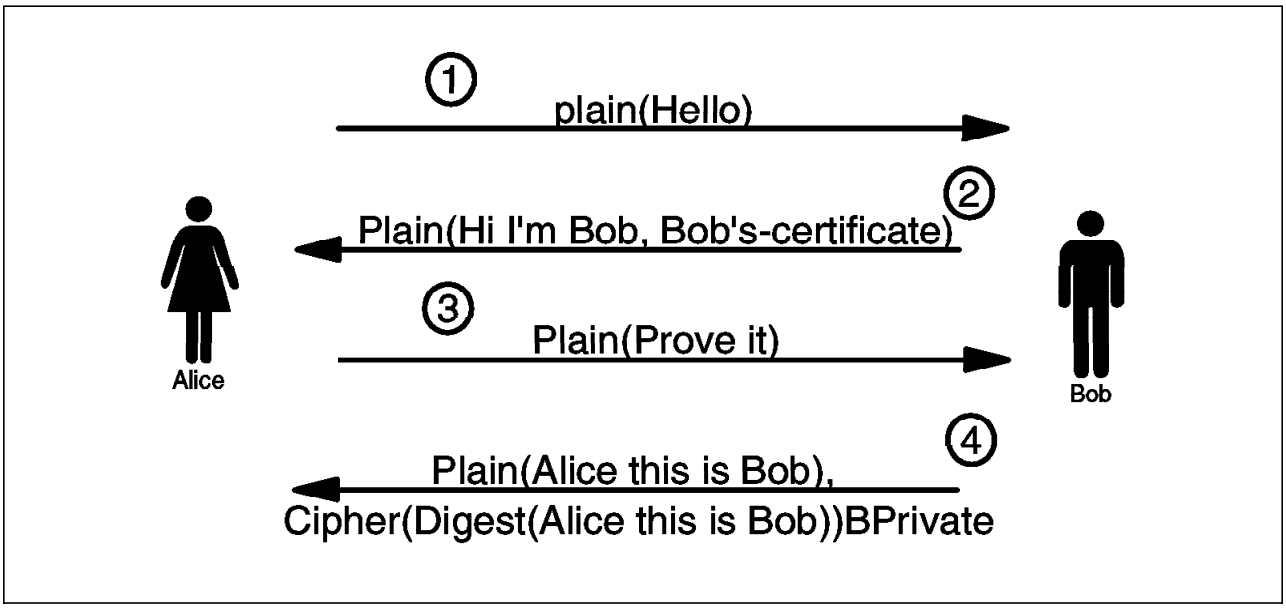


Figure 12. Bob Uses a Digital Certificate to Authenticate Himself

In this example, when Alice receives Bob's first transmission (encircled 2), she checks Bob's certificate Distinguished Name, that the certificate is still valid (has not expired), and that it has been signed by a Certification Authority she trusts. And if Bob is really Bob, then his transmitted public key is valid as well. Alice then challenges Bob (encircled 3) and asks him to send her a *plain(Alice this is Bob), cipher(digest(Alice this is Bob))BPrivate* message. If the message digest of *plain(Alice this is Bob)* is identical to *plain(Alice this is Bob)BPublic*, then Bob has proved that he is acting as himself (encircled 4).

What if Charlie tries to act as Bob?

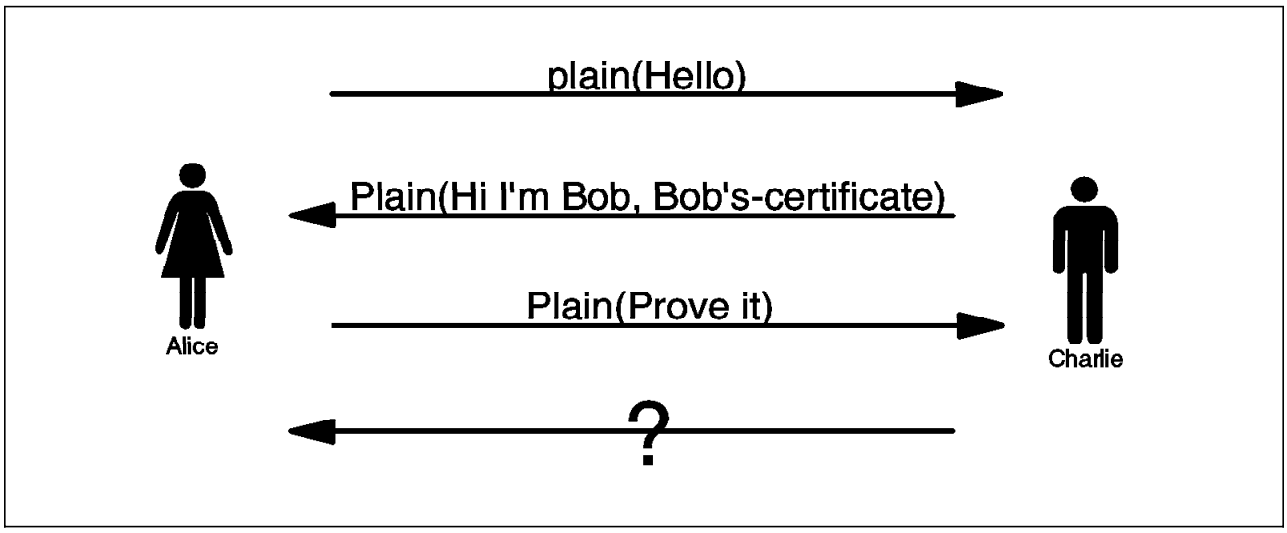


Figure 13. Charlie Tries to Authenticate Himself as Bob

In Figure 13, Charlie has gained access to Bob′s certificate but cannot authenticate himself as Bob because he does not have access to Bob′s Private key (BPrivate).

General TCP/IP Exposure

Messages can always be intercepted and changed simply because everybody using the Internet uses the same protocol suite (TCP/IP) and can listen to, stop, change, and then pass messages on to the original recipient. *The strength of cryptographic system is usually equal to its weakest point.*

In Figure 14, Charlie intercepts and changes a message from Bob to Alice at a stage when Bob has already authenticated himself to Alice. Charlie wants to change the message, but as he only has access to Bob′s public key, he cannot decrypt it, nor can he change it to a meaningful new data string. So he changes it from *secret* to *garble* and passes it on to Alice. When Alice receives the message, she cannot decrypt *garble* into any meaningful message, so the message is thrown away as unreadable.

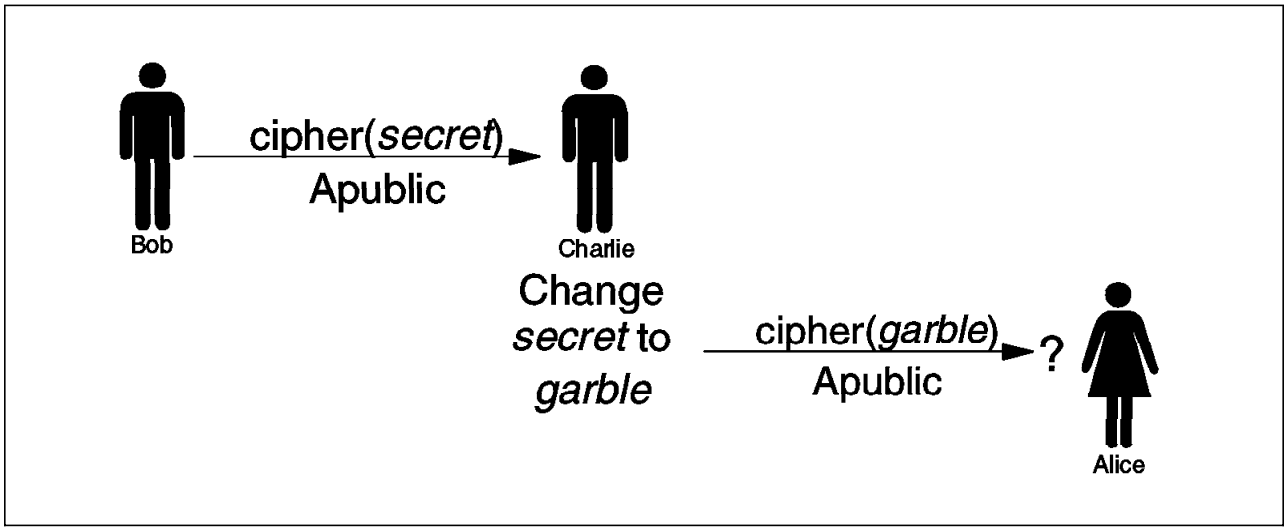


Figure 14. Charlie Garbles a Message from Bob to Alice

A Message Authentication Code (MAC) provides a protection mechanism against this kind of attack. The MAC is simply a message digest that is appended to the message, thus ensuring a way of checking the transmission status when receiving it. As Charlie does not have access to the encryption keys used to generate the MACs, he cannot generate new MACs to authenticate his *garble* messages. When Bob and Alice use the MAC, they know immediately that someone is garbling their messages, and they can stop talking.

4.2.4 Digital Certificate Standard X.509

Certificates have to be standardized in some way and X.509 is one of those standards. X.509 is one of the standards within the X.500 standards hierarchy and describes the contents of a Digital Certificate. An X.509 Digital Certificate consist of these fields:

- Version
- Serial number
- Signature algorithm
- Issuer name
- Validity period
- Subject (user) name
- Subject public key information
- Issuer unique identifier (version 2 and 3 only)
- Subject unique identifier (version 2 and 3 only)
- Extensions (version 3 only)
- Signature on the preceding fields

Having seen how transaction security is achieved through encryption, digital signatures, and digital certificates, next we look at how SSL (Secure Sockets Layer) uses these to establish a secure connection.

4.3 The Secure Sockets Layer Protocol (SSL) and HTTPS

SSL was developed by Netscape Communications Corp. to provide a higher level of security and privacy to TCP applications (for example, HTTP) written to sockets. SSL executes at the transport layer of the TCP/IP stack and it is accessed by HTTPS, which is a unique protocol that combines SSL and HTTP. The HTTPS protocol listens by default on port 443 while the HTTP protocol listens by default on port 80. This means that you can have *both* a secure and a non-secure server running on the same system.

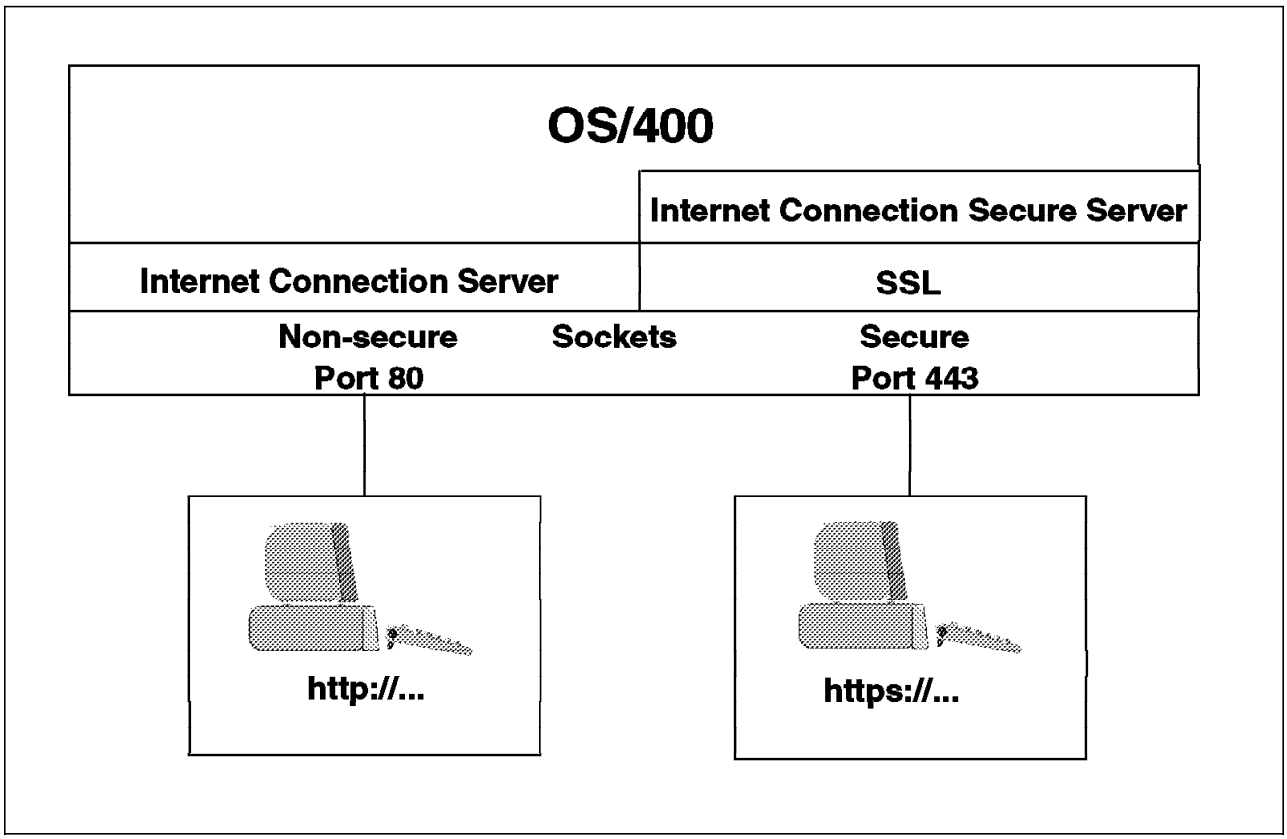


Figure 15. How Internet Connection Server for AS/400 and Internet Connection Secure Server for AS/400 Stacks are Layered in OS/400

Important

Private keys must be stored securely. If compromised, a private key is no longer a secret and privacy is lost with possible forgery and fraud as a result.

4.3.1 The Services Provided by SSL

SSL provides three security services: Message Privacy, Message Integrity, and Mutual Authentication.

Table 1. Services Provided By SSL and Their Paper Document Equivalents

Service	Technology	Protection From	Paper documents
Message Privacy	Encryption	Eavesdroppers	Sealed envelope, Courier delivery
Message Integrity	Message digests	Vandals	Physical Seals
Mutual Authentication	X.509 Certificates	Imposters	Letterhead, Signature, Postmark

Refer to 4.4.2, “ICSS for AS/400 Supported Key Lengths” on page 29 for an overview of standards that SSL supports.

4.3.1.1 Browser Requirements

SSL has a server component and a browser component; to access a secure server, a browser must also support SSL. Without this support, it cannot decrypt the encrypted information and encrypt requests to the server. SSL enabled browsers also have support for digital certificates.

4.3.2 The SSL Handshake

The SSL handshake process is a complex combination of the concepts used in the preceding examples. It all starts when Alice requests an HTTPS session on Bob′s port 443. Alice states that she prefers TripleDES encryption (she could have chosen TripleDES, DES 56 bit, RC2 40/128 bit, or RC4 40/128 bit). Bob sends an OK together with his digital certificate. SSL always negotiates the strongest cipher available that is supported by both parties. Bob′s reply leads to a two-way authentication and verification process using RSA asymmetric encryption (512 bit, 768 bit, or 1024 bit). As soon as Alice is sure about Bob′s identity, she generates a *master session key*. The master session key is used to generate Alice′s read and write encryption keys (TripleDES-168 bit) for the actual data transfer session, and it is also sent to Bob so that he can generate an identical set of read and write keys. Alice′s read key is identical to Bob′s write key and her write key is identical to Bob′s read key. When Bob has generated his session keys, he sends a new verification of Alice′s first message, but now he changes the encryption algorithm from RSA (asymmetric, public key) to TripleDES (symmetric, secret key) as requested by Alice, and uses his session write key to encrypt. From this point, all data from Bob to Alice is encrypted using TripleDES and Bob′s write key and decrypted with Alice′s read key. Data from Alice to Bob is encrypted using Alice′s write key and decrypted using Bob′s read key. Figure 16 on page 26 gives you a schematic impression of the entire process.

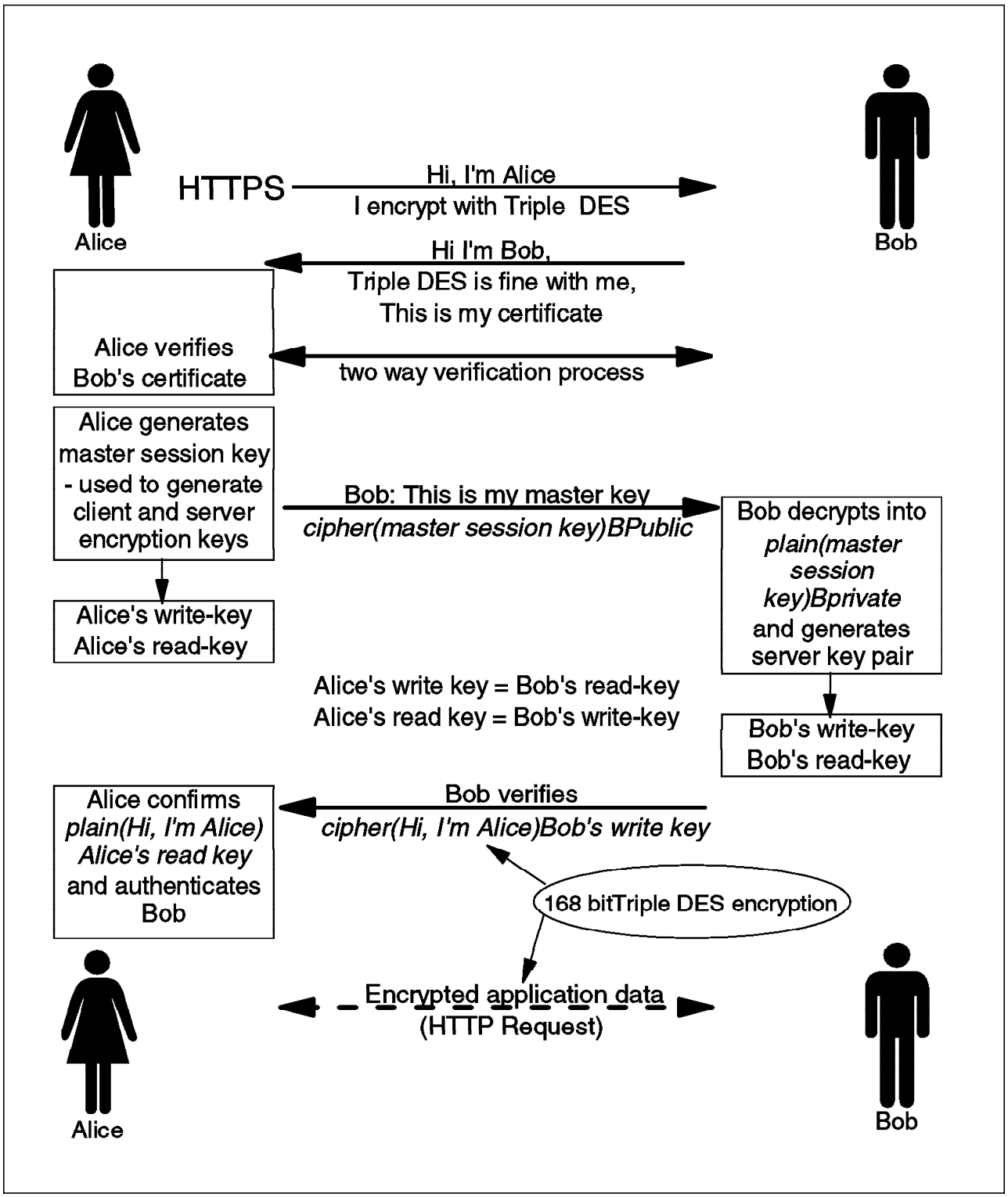


Figure 16. Overview of the SSL Handshake

4.4 The Internet Connection Secure Server for AS/400

Internet Connection Secure Server for AS/400 is required to enable secure Internet serving from the AS/400 system. Internet Connection Secure Server for AS/400 provides secure HTTP transactions with the implementation of several security mechanisms. This section gives a high-level overview of the services offered within Internet Connection Secure Server for AS/400.

Internet Connection Secure Server for AS/400 introduces some new Server Directives to the HTTP configuration file as well as configuration tasks only available from the browser-based configuration. The new server directives are:

Keyfile Set name of the key ring file. The Keyfile is a password protected file where we store the public key, private key, certificate, and trusted root keys. The Keyfile initial configuration file setting is “none” and the program default setting is “none.”

NormalMode
Turn port on or off for HTTP. Use this directive to turn on or off the port specified in the port directive. The NormalMode initial configuration file setting is “none” and the program default setting is “NormalMode on.”

SSLMode Turn port on or off for SSL. Use this directive to turn on or off the port defined by the SSLPort directive. The SSLMode initial configuration file setting is “none” and the program default setting is “SSLMode off.”

SSLPort Set port for SSL security. Use this directive to set the port for SSL security. The server uses this port only for HTTPS requests. The SSLPort initial configuration file setting is “SSLPort 443” and the program default setting is “SSLPort 443.”

Note: If both NormalMode and SSLMode are turned off, the server will *not* start.

Important!

Parts of the SSL/HTTPS code are located within the System Licensed Internal Code (SLIC) area. If, therefore, the SLIC has to be re-loaded for any reason (for example, through a SLIP install), then you have to reload either 5769-NC1 (U.S. and Canada) or 5769-NCE (International) and their PTFs. This can be achieved through the SAVLICPGM/RSTLICPGM commands.

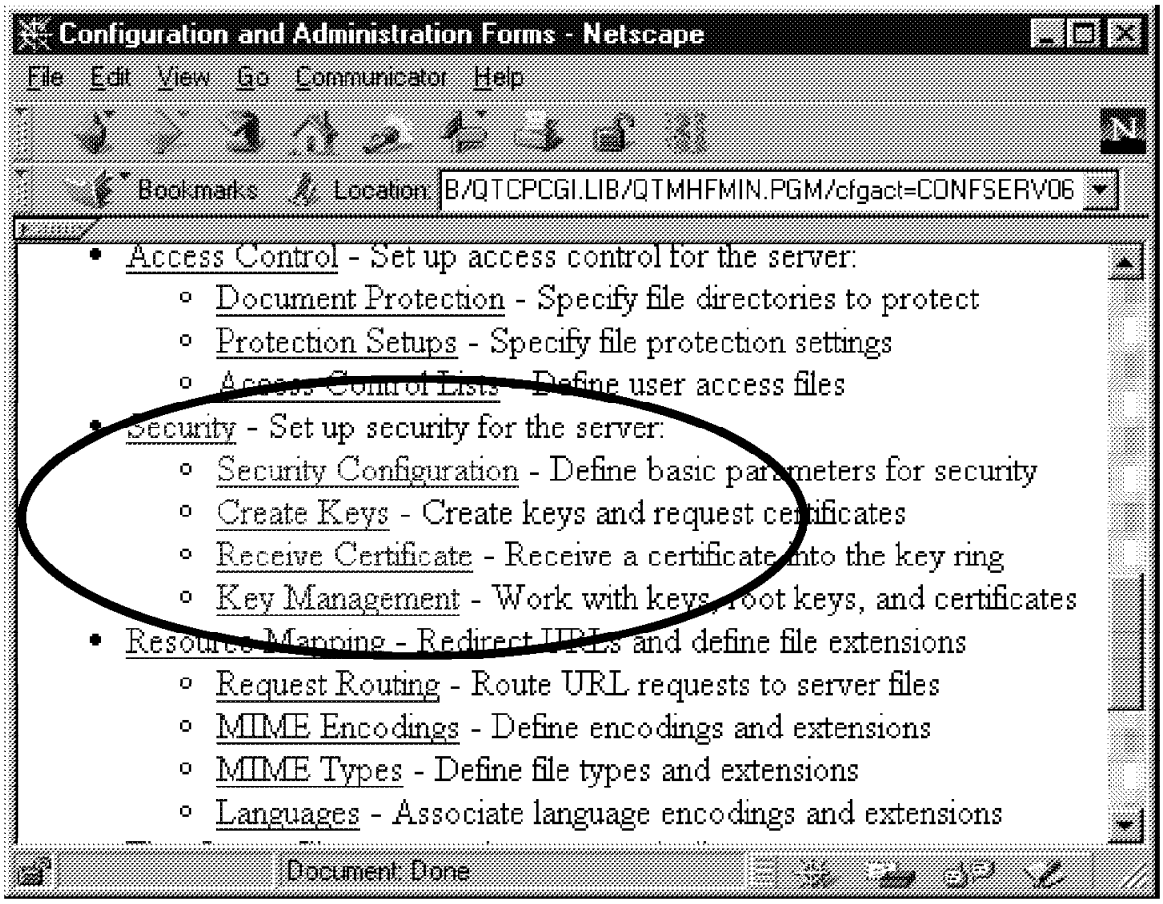


Figure 17. The Security Links of the Configuration and Administration Form

You do not have a secure connection until you do the following steps:

- Install “both” the TCP/IP Connectivity Utilities for AS/400 and either licensed program 5769-NC1 (U.S. and Canada) or 5769-NCE (International).
- Configure the server for SSL.
- Create a key pair for secure network communications.
- Receive a certificate from a certification authority (CA) who is designated as a trusted root on the server.
- Verify that you can establish secure sessions.

Note: If SSLMode is on and you have not completed the previous steps, the server will *not* start.

The Internet Connection Secure Server for AS/400 can enable these services in the AS/400 system:

- Encrypt and check digital certificates in a TCP/IP session using SSL and HTTPS.
- Manage keys, digital certificates, and trusted roots.
- Act as an Certification Authority (CA) in an intranet environment.

4.4.1 ICSS for AS/400 Prerequisites

To enable security, you must install *both* the TCP/IP Connectivity Utilities for AS/400 and either Internet Connection Secure Server for AS/400, 5769-NC1 (U.S. and Canada), or Internet Connection Secure Server for AS/400, 5769-NCE (International). If you have not enabled security, the security functions discussed in this section and in Chapter 9, "Establishing a Secure Connection" on page 159 will not be available.

For Administration and Configuration, you also need a Web browser such as Netscape Navigator or Microsoft Internet Explorer.

4.4.2 ICSS for AS/400 Supported Key Lengths

The U.S. Government regulates products used for encryption. Export of products containing encryption is prohibited unless the key size is limited. The longer the key, the more secure the encryption. As a result of this, two licensed programs are provided: 5769-NC1 (U.S. and Canada) and 5769-NCE (International).

As established in 4.3.2, "The SSL Handshake" on page 25, SSL uses two algorithms for encryption. RSA is used for encryption during the SSL handshake and the exchange of session keys. RC2, RC4, DES, or TripleDES is used for encryption during the actual data transmission.

RSA encryption key sizes for 5769-NC1 (U.S. and Canada):

.

- Generates keys for RSA encryption from 508 bits to 1024 bits.
- Encrypt session keys with keys from 508 bits to 1024 bits.
- Sign certificates with keys from 508 bits to 1024 bits.
- Check signatures with keys from 508 bits to 1024 bits.

RSA key sizes for 5769-NCE (International):

- Generates keys for RSA encryption from 508 bits to 512 bits.
- Encrypt session keys with keys from 508 bits to 512 bits.
- Sign certificates with keys from 508 bits to 512 bits.
- Check signatures with keys from 508 bits to 1024 bits.

The SSL modes for the U.S./Canadian version (5769-NC1 (U.S. and Canada)) are:

- RC4 128 bit
- RC2 128 bit
- DES 56 bit
- Triple DES (EDE) 168 bit

Note: The server uses RC4 export (128 bit; 40 secret) or RC2 export (128 bit; 40 secret) modes when communicating with an export version of an SSL client.

The SSL modes for the export version (5769-NCE (International)) are:

- RC4 export (128 bit; 40 secret)
- RC2 export (128 bit; 40 secret)

Note: The key field length for RC4 and RC2 is *always* 128 bits. For 5769-NC1 (U.S. and Canada), all 128 bits of the encryption algorithm are secret. For 5769-NCE (International), only 40 bits of the encryption algorithm are secret; the remaining 88 bits are known.

Internet Connection Secure Server for AS/400 supports the following standards:

- Netscape SSL V2 and V3 specifications (defacto SSL standard)
- Certificate processing:

  X.509 certificates
  ASN.1 DER encoding
  RFC-1424 PEM-based certificate requests

- RSA PKCS (Public Key Cryptography Standards):

  PKCS-1: RSA encryption
  PKCS-5: Encrypt RSA private key with password
  PKCS-7: Cryptographic message syntax
  PKCS-8: RSA private-key information syntax

Internet Connection Secure Server for AS/400 implements the following SSL functions:

Encryption	DES, TripleDES, RC2, RC4
Message Digest	MD2, MD5, SHA-1
Public key encr.	RSA
Digital Signature	RSA w/MD2, RSA w/MD5

4.4.3 A Few Words on Performance

The implementation and activation of Internet Connection Secure Server for AS/400 has a performance impact on the transactions being transmitted. This is caused by the simple fact that encryption is an extra processing step that is performed for secure sessions and not performed for non-secure sessions. This impact is a function of the complexity of the encryption algorithm and processing time at both the browser and server to encrypt/decrypt.

This performance impact is not unique to the AS/400 system, but is a characteristic of all SSL (or any other encryption) mechanisms.

The first page is usually the longest because of the session setup information that is exchanged between the server and the browser.

4.5 A List of SSL Related Terms

Armor password: The password used when exporting a key pair or certificate to another computer. The export file will be encrypted using this password.

Asymmetric Keys: In secure communication, the two keys in a key pair. The key pair consists of a private key and a public key. The keys are called asymmetric because one part (private/public) can only decrypt what has been encrypted with the other (public/private) and the other way around.

Authentication (1): Verification that a message has not been altered or corrupted.

Authentication (2): A process used to verify the user of an information system or protected resource.

Certificate: A digital document that binds a public key to the identity of the certificate owner, thereby enabling the certificate owner to be authenticated. A certificate is issued by a Certification Authority (CA).

Certification Authority (CA): An organization that issues and signs certificates. The CA authenticates the certificate owner's identity and the services the owner is authorized to use.

Ciphertext: Encrypted data, the result of the encryption process.

Cleartext: See Plaintext.

Decryption: The transformation of encrypted data (or ciphertext) into plaintext.

Digital Signature: Information that is encrypted with the entity's private key and is appended to a message to assure the recipient of the authenticity and integrity of the message. The digital signature proves that the message was signed by the entity that owns, or has access to, the private key.

Distinguished name: The name and address used to identify the owner and issuer field within a digital certificate.

Encryption: The transformation of data into a form unreadable by anyone without the secret decryption key.

Export(ing) and import(ing) keys: If you need to transfer a key pair or certificate to another computer, you can export it to a file. On the other computer, you can import it into a key ring.

Hash function: See Message Digest Function.

Hypertext Transfer Protocol Secure (HTTPS): A TCP/IP protocol that is used by World Wide Web servers and Web browsers to transfer and display hypermedia documents securely across the Internet.

ICSS for AS/400 files: ICSS for AS/400 files: Keyring files (.KYR), password files (.STH) and request and certificate files (.TXT).

Import keys: See "Export and import keys."

Key: An algorithmic pattern used by a sender to encrypt messages and by the recipient to decrypt messages.

Key pair: A public and a private key.

Key ring: A file that contains public keys, private keys, trusted roots, and certificates.

Message Authentication Code (MAC): A code generated using a message digest algorithm. A MAC is generated by inputting some of the data from the current message and a secret key to the message digest algorithm.

Message Digest Function: A function that generates a fixed-length message digest from a variable-length string input. This function is also known as a Hash function. The message digest function uses a message digest algorithm to achieve this.

Plaintext: Readable text (unscrambled, not encrypted).

Private key: An algorithmic pattern used to encrypt messages that can be decrypted only by the corresponding public key. The Private Key is also used to decrypt messages encrypted by the corresponding public key. The Private Key is kept on the user′s system and is protected by a password.

Public key: An algorithmic pattern used to decrypt messages that were encrypted by the corresponding private key. A Public Key is also used to encrypt messages that can only be decrypted by the corresponding private key. Users broadcast their Public Keys to everyone with whom they must exchange encrypted messages.

Public key cryptography: A method of cryptography that depends on a matched pair of keys. Information encrypted with one key can be decrypted only with the other key in the pair. One of the keys is made public; the other key is kept private. This method allows secure communication between an individual entity (such as a merchant) and any other entity who obtains the public key (such as consumers).

Session keys: Keys used within SSL for symmetric encryption during the data transmissions. Session keys are negotiated during the SSL handshake. They are unique to every session and are not used again.

Secure Sockets Layer (SSL): A popular security scheme developed by Netscape Communications Corp. and RSA Data Security, Inc. that allows the client to authenticate the server and all the data and requests to be encrypted. The URL of a secure server protected by SSL begins with https rather than http.

Trusted Root: The public key and associated distinguished name of a certificate authority.

Trusted Third Party (TTP): An external organization that is trusted as an independent part in certain activities. A certification authority is a TTP, but there are also TTPs that do not deal with certification or Internet security.

Chapter 5. Web Browser Configuration Interface

One of the differences between ICS for AS/400 and the HTTP server shipped with the AS/400 TCP/IP Connectivity Utilities V3R2 and V3R7 is that you can configure ICS for AS/400 using a Web browser. To do this, we use another new feature called the *administration server*. As mentioned previously, ICS for AS/400 allows you to define multiple server instances. The TCP/IP Utilities includes a server instance called *ADMIN*. The ADMIN server is shipped pre-configured. The home page served by the ADMIN server is referred to as the AS/400 Tasks page. From this page, you can configure, among others, Internet Connection Server for AS/400 and Internet Connection Secure Server for AS/400. In this chapter, we explain how to start the ADMIN server, how to start the configuration process, and how the configuration interface looks.

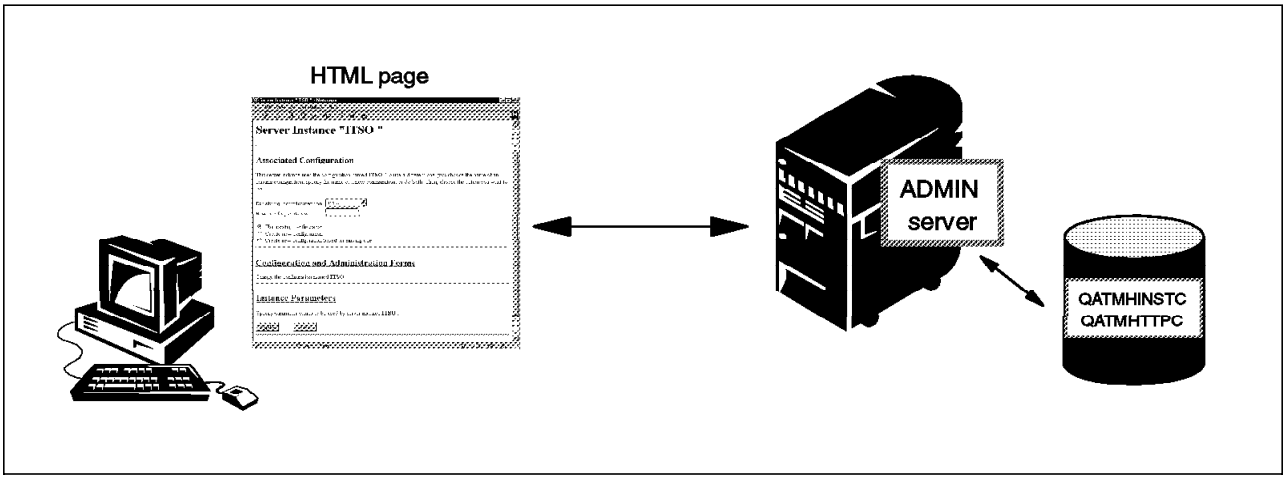


Figure 18. ADMIN Server

The Web browser configuration interface is new but much of the underlying structure is still the same as in V3R2 and V3R7. The server configurations are still stored as file members in the QATMHTTPC file in the QUSRSYS library (except for the ADMIN server). To accommodate the server instances, the QATMHINSTC file has been added to the QUSRSYS library. It contains the server instance information for all servers except the ADMIN server.

Configuration files

Configuration files QATMHTTPC and QATMHINSTC are used internally by the system. The configuration interface to these files should be either through a Web browser (the preferred interface) or through WRKHTTPCFG. It should not be necessary to manipulate these files directly.

Note: You can still configure ICS for AS/400 using the old 5250 terminal based methods. For more information, see *Cool Title About the AS/400 and Internet*, SG24-4815.

5.1 Administration Server

To start the ADMIN server, enter:

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

Important Note about Special Authorities

To perform the configuration or administration of the server, you must have a valid AS/400 user profile. In addition, these special authorities are required for the following tasks:

- **5769-TC1 Installation** - must have *ALLOBJ special authority or authority to the Restore Licensed Program (RSTLICPGM) command object.
- **Configuration or Administration** - must have:
  - *IOSYSCFG special authority
  - *CHANGE authority to library QUSRSYS
  - *ALL authority to file QUSRSYS/QATMHTTPA
  - *ALL authority to file QUSRSYS/QATMHTTPC
  - *ALL authority to file QUSRSYS/QATMHINSTA
  - *ALL authority to file QUSRSYS/QATMHINSTC
- **Start and stop** - must have *IOSYSCFG special authority and *USE authority to the Start TCP Server (STRTCPSVR) command and End TCP Server (ENDTCPSVR) command objects.

We used user profiles that had *SECOFR user class and the special authorities that come with that user class to avoid difficulties.

If you are doing server configuration over the Internet, you may want to use a secure connection. The configuration process involves sending a user profile name and a password over the connection and it is advisable not to do this over a non-secure connection. When configuring over an intranet, these precautions are perhaps not necessary. To use the secure connection, you need to perform the following steps:

1. Install the Internet Connection Secure Server for AS/400 (U.S.), 5769-NC1, or Internet Connection Secure Server for AS/400 (International) 5769-NCE.
2. Through the administration server, configure the secure sockets layer (SSL) components for the ADMIN server.

For more information about the secure connection and its configuration, see Chapter 4, “AS/400 Internet Connection Secure Server” on page 15 and Chapter 9, “Establishing a Secure Connection” on page 159. For more information on customizing the ADMIN server instance, see the *Webmaster's Guide*, GC41-5434.

Note: If the ADMIN server's certificate or key ring password expires, the ADMIN server will not start with SSLMode on. You need to use the WRKHTTPCFG (*ADMIN) command to turn SSLMode off; then use the ADMIN server in normal mode to access the configuration and administration forms to change the key ring password or request a new certificate.

Any changes to the ADMIN server (for example, the addition of an SSL configuration) are placed in QUSRSYS/QATMHTTPA.ADMIN. This file is read by the ADMIN server instance prior to the internal configuration file being read.

5.2 Using the Browser Interface to Configure ICS for AS/400

Before trying to access the ADMIN server, it is good idea to verify that the server is running. To do this, use the WRKACTJOB JOB(ADMIN) command (see Figure 19).

```
                              Work with Active Jobs                    SYSTEM01
                                                                  11/13/97  10:48:57
CPU %:    1.7     Elapsed time:   00:04:00     Active jobs:   235
Opt  Subsystem/Job  User        Type  CPU %  Function        Status
__     ADMIN        QTMHHTTP    BCH      .0  PGM-QTMHHTTP     TIMW
__     ADMIN        QTMHHTTP    BCI      .0  PGM-QYUNLANG     DEQW
__     ADMIN        QTMHHTTP    BCI      .0                   SELW
__     ADMIN        QTMHHTTP    BCI      .0                   SELW
__     ADMIN        QTMHHTTP    BCI      .0                   DEQW
__     ADMIN        QTMHHTTP    BCI      .0                   DEQW
__     ADMIN        QTMHHTTP    BCI      .0                   DEQW
                                                                          Bottom
===>
F21=Display instructions/keys
```

Figure 19. Result of WRKACTJOB JOB(ADMIN)

Once the ADMIN server is up and running, you can start configuring the server. To do this, you need a Web browser that supports the HTTP 1.0 protocol such as Microsoft Internet Explorer for Windows 95 or Netscape Navigator for Windows 95, AIX, or OS/2. The browser should also support frames. If you want to view the online help on the administration server, your browser must be set so that JavaScript is enabled.

Browser Cache and Proxies

Before using a browser to configure ICS for AS/400 or ICSS for AS/400, disable the browser cache and proxies. Because the cache is disabled, avoid using the browser *Back* and *Forward* buttons.

To get to the AS/400 Tasks page, point your browser to one of the following locations:

```
http://your.server.name:2001
https://your.server.name:2010
```

The port 2001 has been reserved for a non-secure connection and port 2010 is for a secure connection. Use the latter only if you have defined the ADMIN server for a secure connection.

Next, you are asked for a valid AS/400 user profile and password (see Figure 20 on page 36).

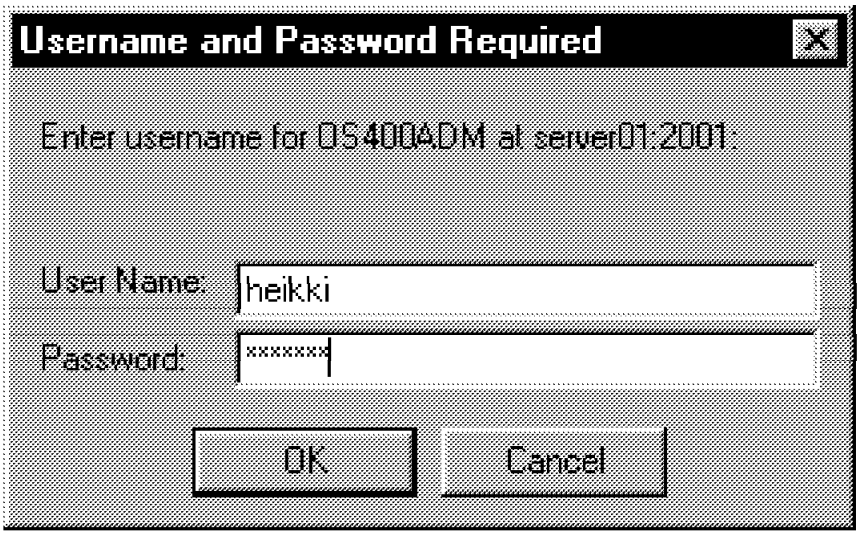


Figure 20. Username and Password Prompt

After the validation of these, the AS/400 Tasks page is shown (see Figure 21).

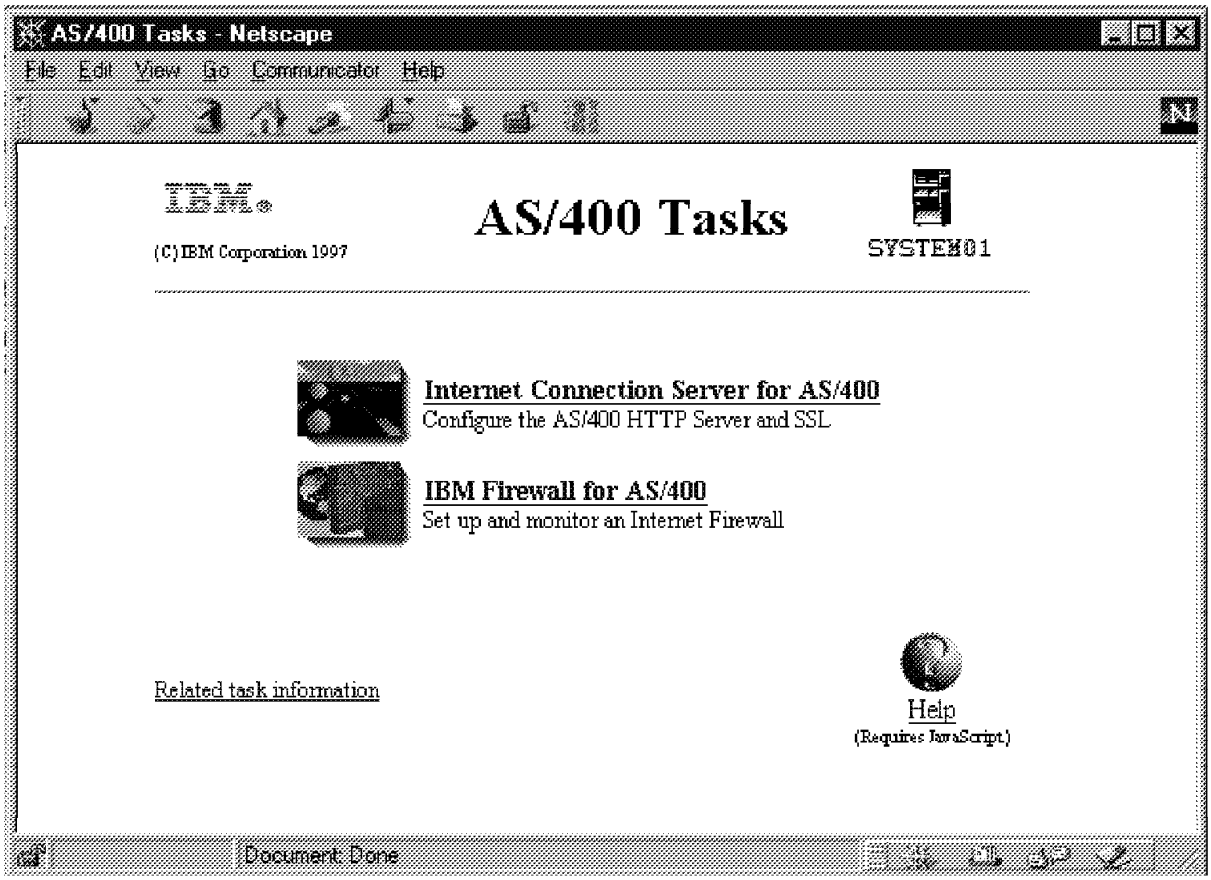


Figure 21. AS/400 Tasks

The AS/400 Tasks page contains links to products that are configurable through a browser. It also has online help that requires JavaScript and a link *Related task information* page. The Related task information page is in the Internet and contains links to different AS/400 information and publication sites.

After selecting the Internet Connection Server for AS/400 link, the following page is shown:

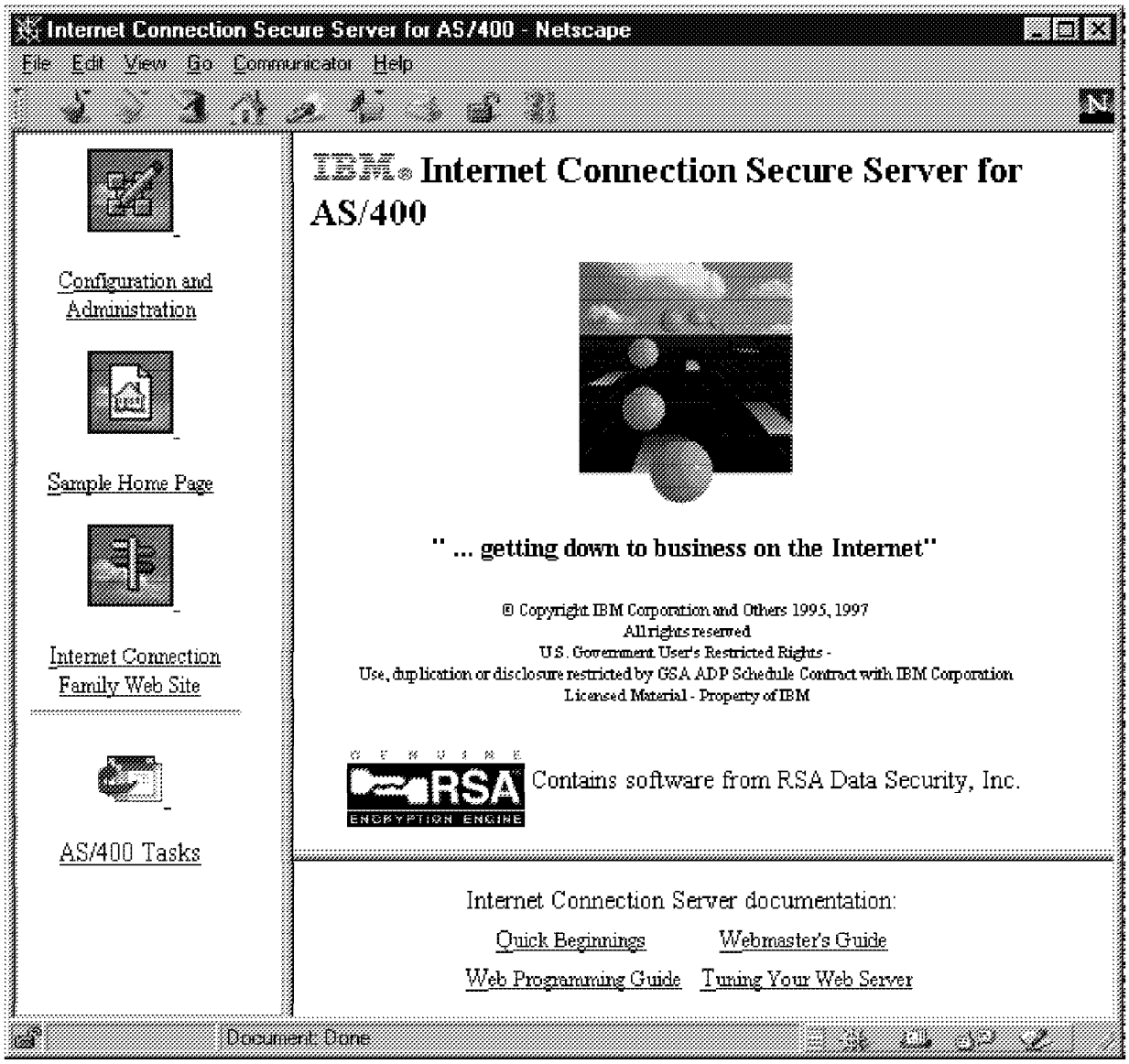


Figure 22. ICS for AS/400 Administration Front Page

This page has links to the *Configuration and Administration* page and *Sample Home Page* on the local server. The *Internet Connection Family Web Site* is located in the Internet. *AS/400 Tasks* returns you to the previous page. *Quick Beginning* and *Webmaster's Guide* are local links and the rest are located in the Internet.

After selecting the **Configuration and Administration** link, the *General Configuration and Administration* page is shown (see Figure 23 on page 38).

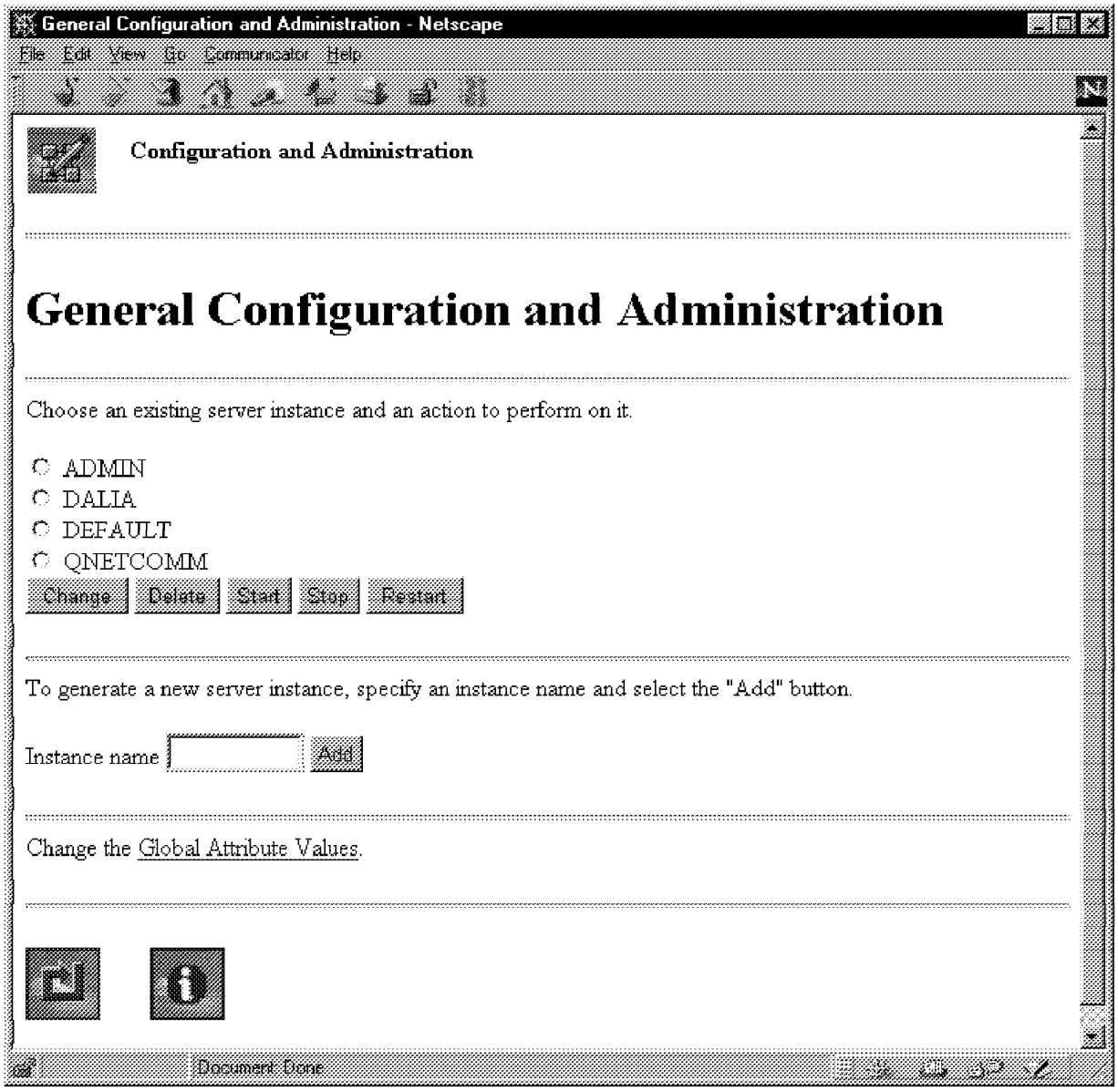


Figure 23. General Configuration and Administration Page

From this page, select the server instance you want to configure. Press the **Change** button and from the next page, select the **Configuration and Administration Forms** link. The *Configuration and Administration Forms* page is shown. This is the main menu for configuring ICS for AS/400. To create a new server instance, enter a name for the new instance and press Add.

Note: You can also delete, start, stop, and restart server instances from this page.

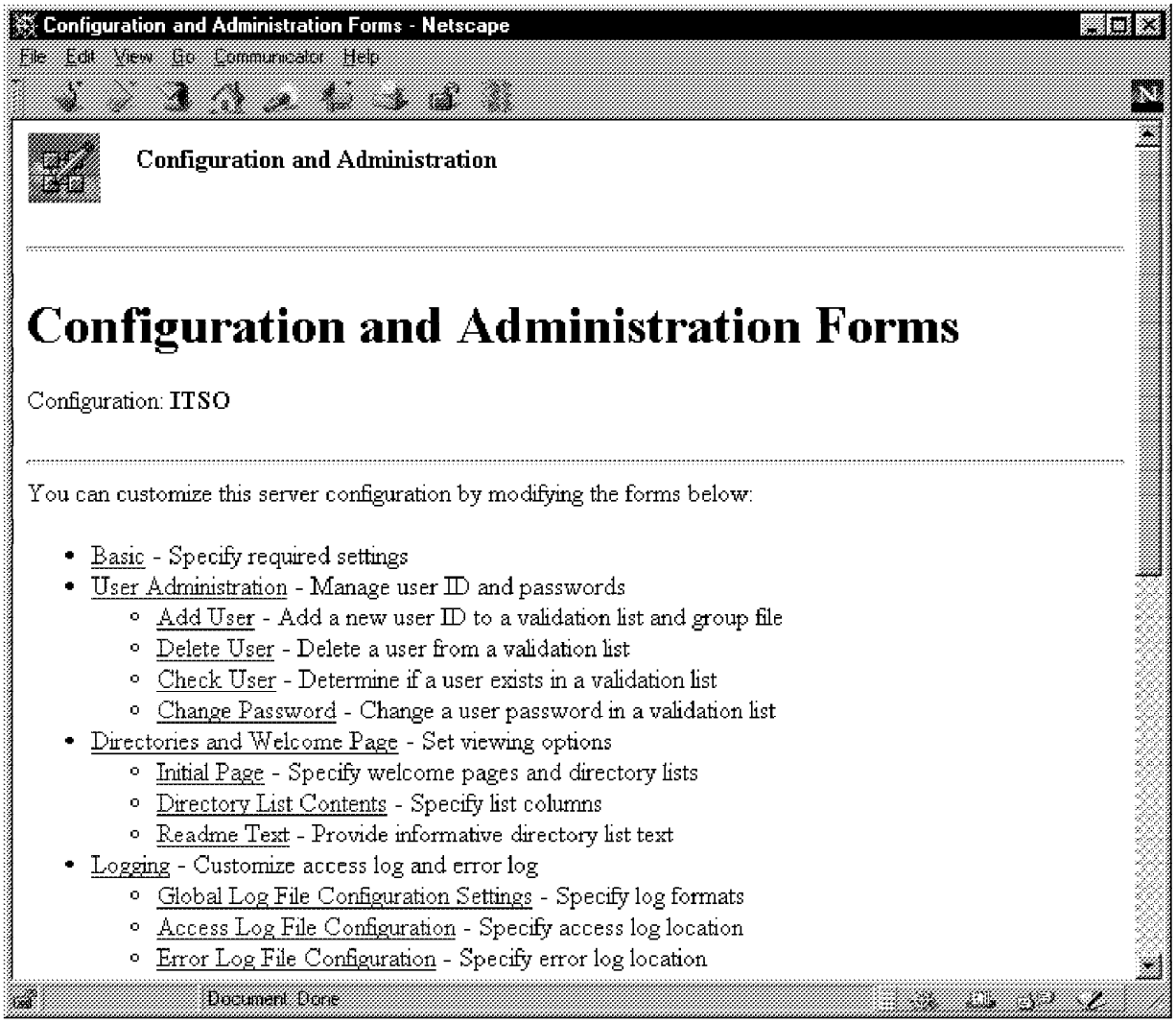


Figure 24. Configuration and Administration Forms Page

From the Configuration and Administration Forms page, you can get to all of the configuration options for a server.

As an example of the configuration interface, we have selected the README Text configuration option (see Figure 25).

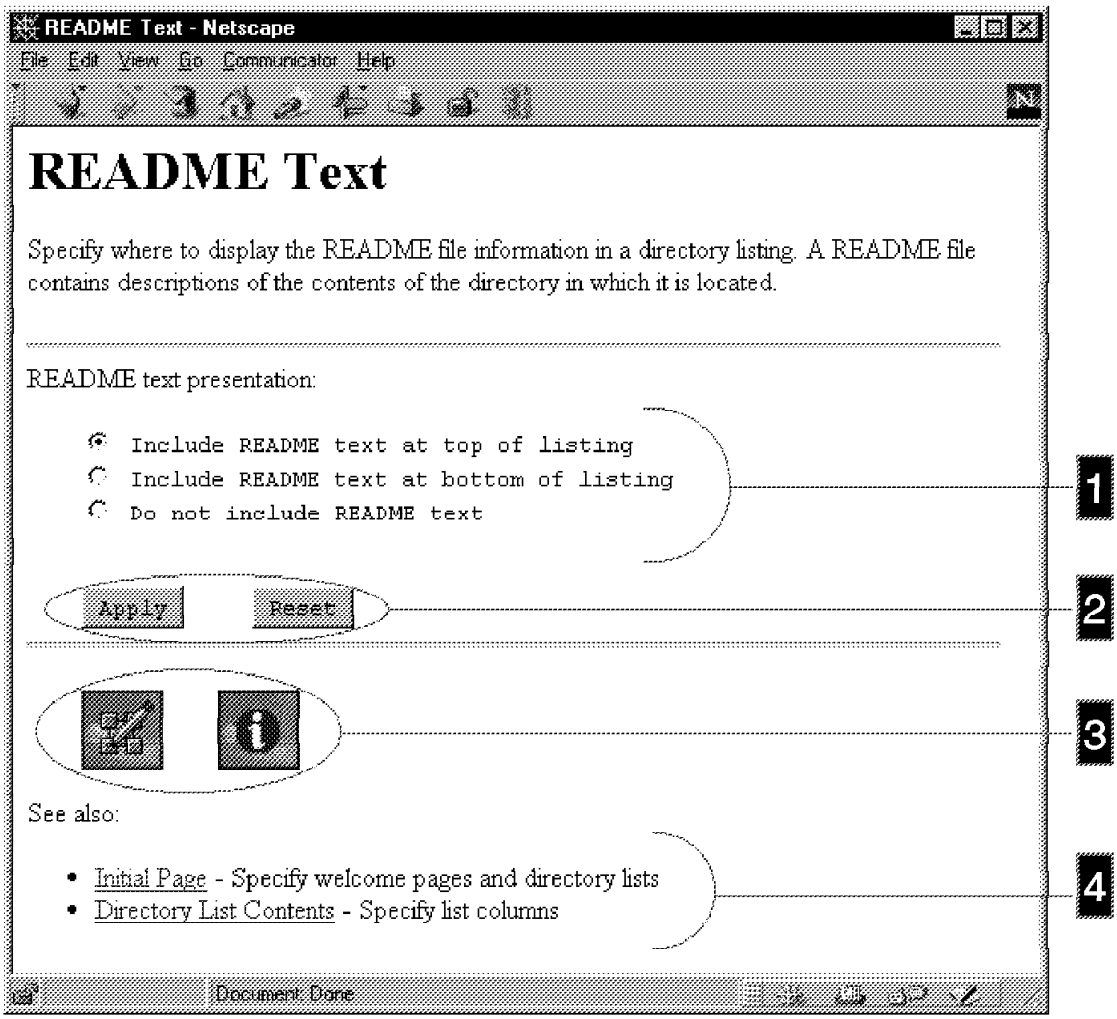


Figure 25. Example Configuration Page

An options page may have a settings area (**1**), buttons area (**2**), icons area (**3**), and related options area (**4**). The settings area may show present settings. The settings may be entered by radio buttons, pull-down menus, check boxes, or input fields. The Reset button resets the settings. The left icon returns to Configuration and Administration Forms page, and the right icon shows a help page related to the options.

After you have filled in the options and pressed the Apply button, a *Confirmation* page or a *Configuration Error* page is shown (see Figure 26 on page 41 and Figure 27 on page 42).

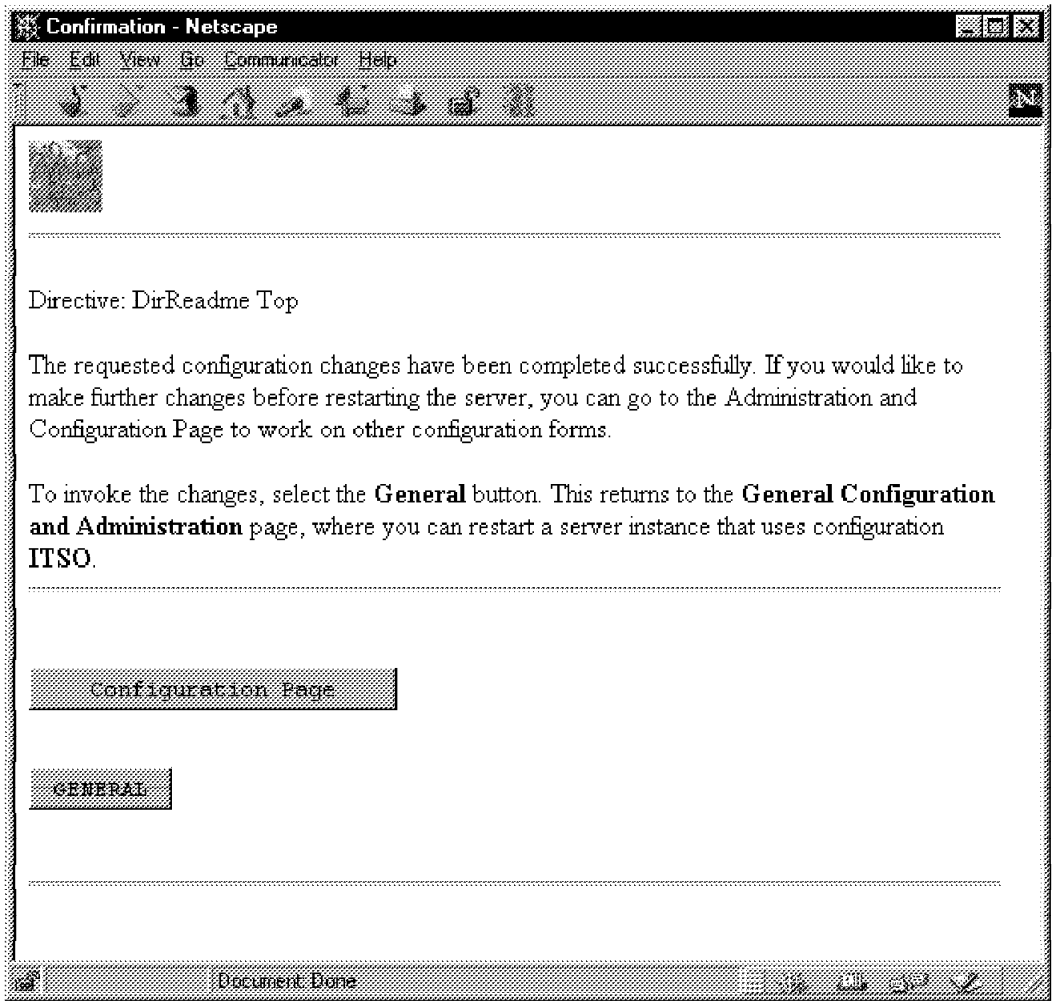


Figure 26. Confirmation Page

The Confirmation Page typically shows the generated configuration directives. When you press the Configuration Page button, you return to the Configuration and Administration Forms page. Pressing the General button returns you to General Configuration and Administration page.

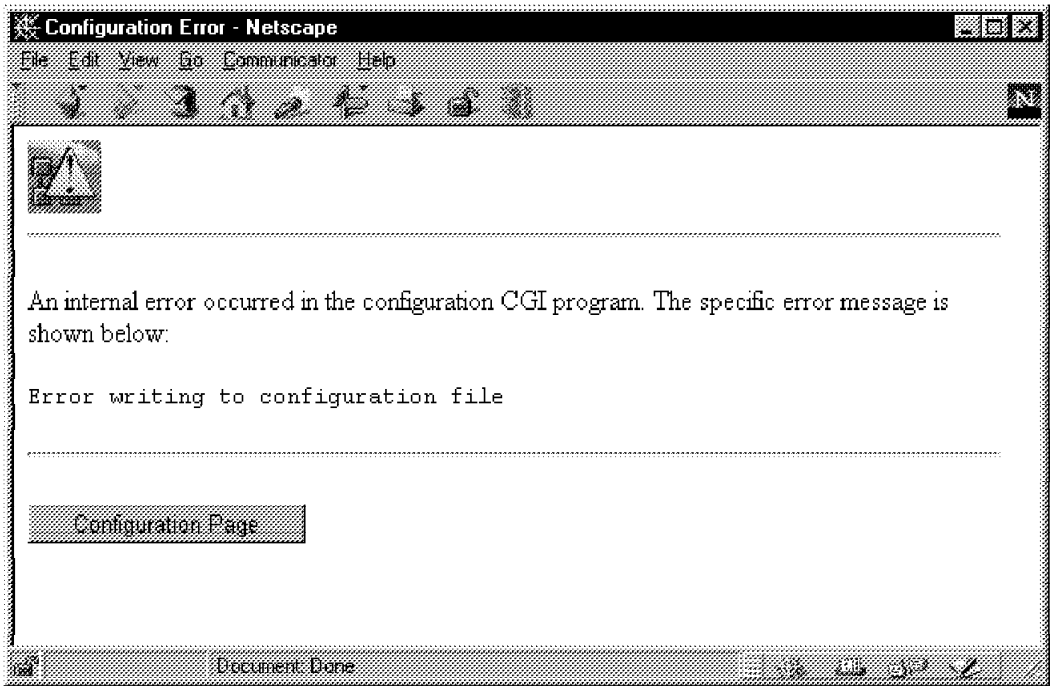


Figure 27. Configuration Error Page

The Configuration Error Page informs you that the update failed for some reason. The error messages are usually quite cryptic and may not tell you much about the actual reason for the error. It may sometimes help to change the ADMIN server to run only one server job so that you can find the error messages from the server's job log. (For information on changing the ADMIN server to run only one job, see Section 6.10.1, "Jobs" on page 99)

Note: The error in Figure 27 was generated by displaying the configuration file member on a 5250 terminal while doing the update with the browser.

Chapter 6. Basic Internet Connection Server Configuration

In this chapter, we look at some general ICS for AS/400 planning considerations and take a look at basic server configuration using the ADMIN server configuration forms.

For information on access control considerations and configuration, see Chapter 8, “Protecting Server Resources” on page 121.

For information on ICSS for AS/400 specific considerations and configuration, see Chapter 9, “Establishing a Secure Connection” on page 159.

For information on using the ADMIN server configuration interface, see Chapter 5, “Web Browser Configuration Interface” on page 33.

6.1 Planning Considerations

Before starting the server configuration, you need to do some planning. You need to decide what kind of server you are setting up, ICS for AS/400 or ICSS for AS/400. You also need to decide from which AS/400 file system you are going to do the Web serving. You might have a need for multiple server instances, or one server instance serving different files to different subnets might be enough. These issues are addressed in the following sections. Chapter 7, “How to Get ICS for AS/400 Server Up and Running” on page 103 provides the minimum steps you need to perform to get a server up and running.

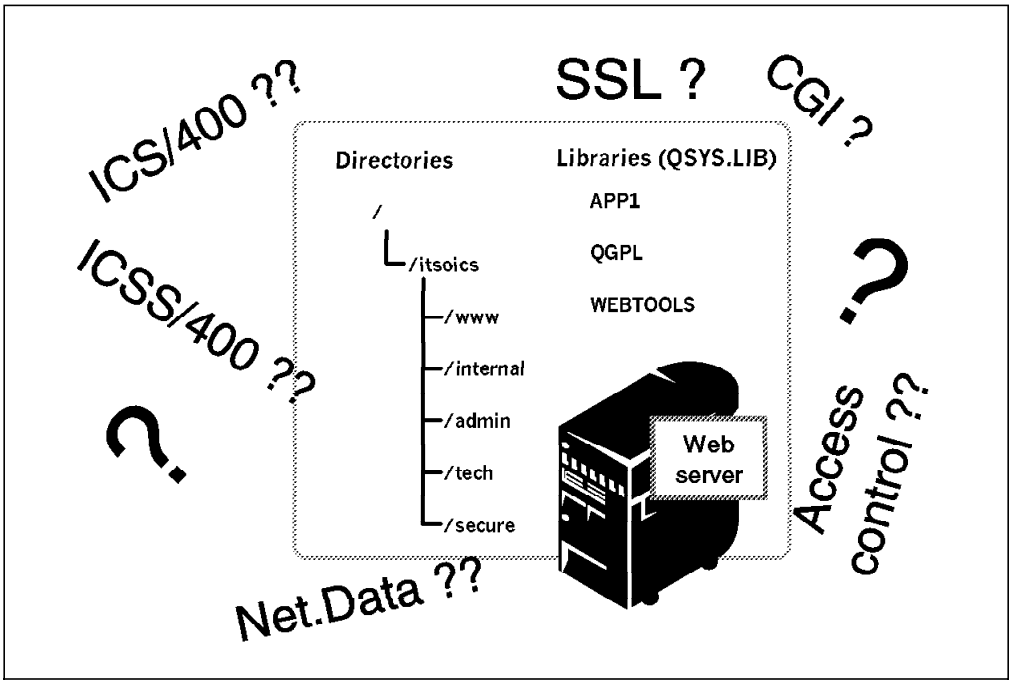


Figure 28. To Serve or Not To Serve?

6.1.1 Basic Planning

In this section we cover some of the planning aspects that should be considered prior to implementing a Web server. We do not, however, cover all the aspects that should be considered. One area that is not covered in detail is network security. For further reading on network security, see *AS/400 Internet Security: Protecting your AS/400 from HARM on the Internet*, SG24-4929 and *AS/400 and the Internet*, G325-6321.

6.1.1.1 Secure or Non-secure?

ICSS for AS/400 provides a secure, encrypted, connection between a browser and the AS/400 Web server. If you are setting up an intranet server, there may be no requirements for ICSS for AS/400. But when connecting to the Internet, it is important that consideration be taken for the need for a secure, encrypted session. This is especially the case when serving dynamic pages (using CGI, Net.Data, or Java applications) that contain customer specific, and confidential data. The static HTML pages, however, may not need the encryption protection provided by the ICSS for AS/400.

The performance overhead for a secure transaction is a fixed cost in terms of additional CPU time and network traffic. For light transactions (for example, static pages), serving secure pages may double the CPU time requirements (cut capacity in half). For heavier transactions (for example, using Net.Data for SQL database access), the overhead is much less percentage wise. The following graph (see Figure 29) shows the relative overhead for secure transactions.

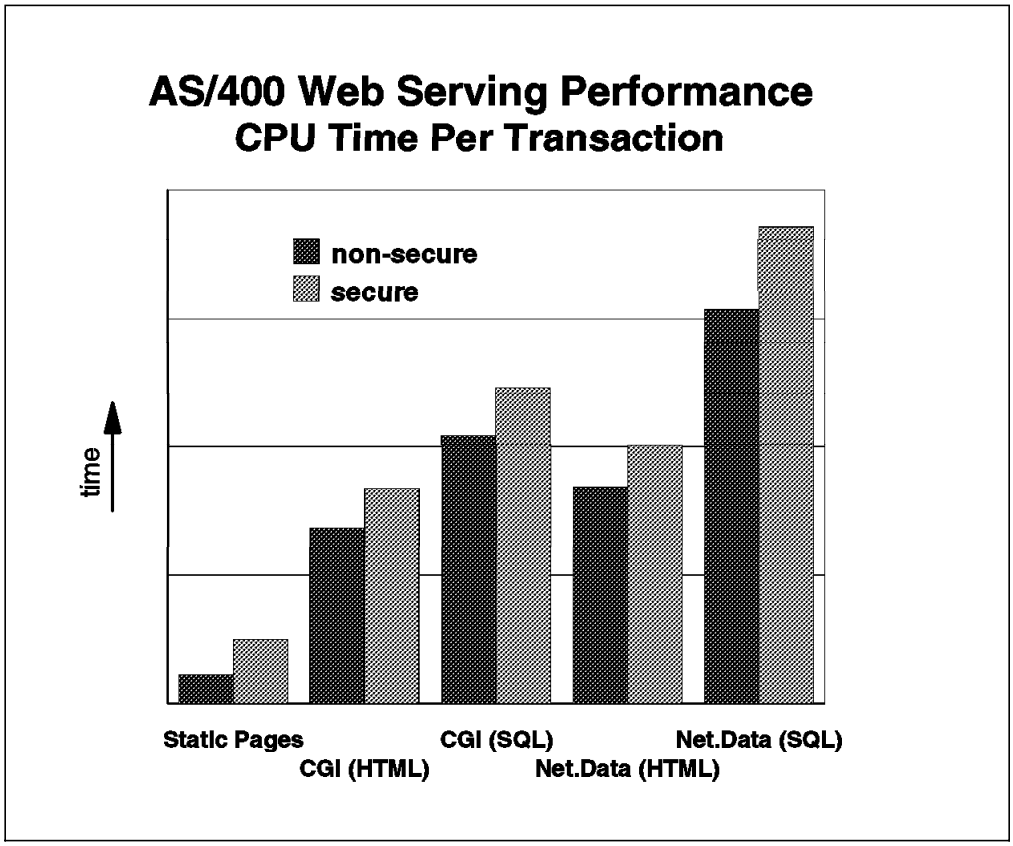


Figure 29. CPU Time per Transaction

As you can see, the impact is not as large for the pages for which a secure connection is most needed. We should use encryption only for the pages that

really need the protection and secrecy. For more information about setting up ICSS for AS/400, see Chapter 9, “Establishing a Secure Connection” on page 159.

6.1.1.2 File System, Libraries, and Directories

Another thing we have to decide is the AS/400 file system from which you do the serving. As we have mentioned earlier, the ICS for AS/400 and ICSS for AS/400 can serve files from following file systems:

- ″root″
- QOpenSys
- QLANSrv
- QOpt
- QFileSvr.400
- QDLS
- QSYS.LIB

Which AS/400 file system works best? There is no simple answer to this question but we can give you a few things to think about. The good news is that you are not stuck with any one file system. You can use a combination of file systems to apply the best solution to each situation. Some of the primary considerations are:

- **Performance** is usually a primary consideration. Tests show that, in a typical configuration, serving from the root directory is the fastest. Here are the approximate relative numbers:
  - ″root″ = 1 (fastest)
  - QDLS = .75
  - QSYS.LIB = .45
  - QLANSrv = .22

  QOpenSys performs similar to the IFS root system.
- **Environment** is also important. How do you intend to create and maintain the HTML files? With Client Access or FTP, it is easy to move files from your PC to a directory in the root or QDLS. You can then take advantage of PC client tools for maintaining HTML files. On the other hand, you can use SEU and create the files right on a ″green screen″ terminal. It is not a lot of fun but it allows you to utilize data entry people that are familiar with editing file members.
- **Flexibility** is an issue if you want to transport the pages from one platform to another. A typical situation is that somebody wants to load Web pages on a laptop and show it to customers or some other people. If you keep file names to eight characters with three-character extensions, use relative path names for links, and store the files in a directory structure under the Integrated File System root, you can easily replicate the structure on a PC.
- **Applications** may also affect your decision. For example, if you have a CGI application that interacts with an HTML page, you may want to keep that page in the same library as the program. This keeps the entire application in one place. Remember that the CGI programs must reside in the QSYS.LIB file system.

In general, HTML files stored in a directory structure under the Integrated File System root perform the best and are easy to maintain (check links, make global changes).

After you have decided in which of the file systems the files are going to reside, the next thing to decide is directory and library structures. Too deep a directory structure may have a performance impact because of multiple authority lookups. Also it is a good idea is to plan a good file path replacement scheme so as to hide the actual directory structure from would-be hackers. We are using the following scheme in our examples:

Table 2. An Example of Replacement File Path Scheme

Request Template	**Replacement File Path**
/internal/*	/itsoics/internal/*
/admin/*	/itsoics/admin/*
/tech/*	/itsoics/tech/*
/secure/*	/itsoics/secure/*
/www/*	/itsoics/www/*

For more information on file path replacement setup, see Section 6.6.1, “Request Routing” on page 80.

Remember also to plan an access control scheme for the files. If you are going to use ACL files to limit access to specific files on a protected directory, then the files must reside in one of the following file systems:

'root'
QLanSrvr
QOpenSys
QDLS

For more information on protecting the server resources, see Chapter 8, “Protecting Server Resources” on page 121.

6.1.1.3 Multiple Instances and Multiple IP Addresses

You can install the server on a machine with multiple network connections. If you do, you can configure the server to serve different files based on the IP address of the network connection a request comes in on. This can be of particular value if you are an Internet service provider and want to use one server to provide Web sites for multiple customers.

For example, you may have two customers (companyA and companyB), both of whom want to make information about their companies available on the World Wide Web. However, the expected number of requests for the information is not great enough that you want to have a separate machine for each customer.

If the machine has two network connections, you can run just one instance of the server and assign each customer to a different IP address. For each IP address, define a different host name. CompanyA can be www.companyA.no on IP address 9.67.106.79 and companyB can be www.companyB.fi on IP address 9.83.100.45. You can then configure the server to serve a different set of information depending on the IP address the request comes in on. Because the server can accept requests from the default port of each network connection, requests to either host name do not require a port number.

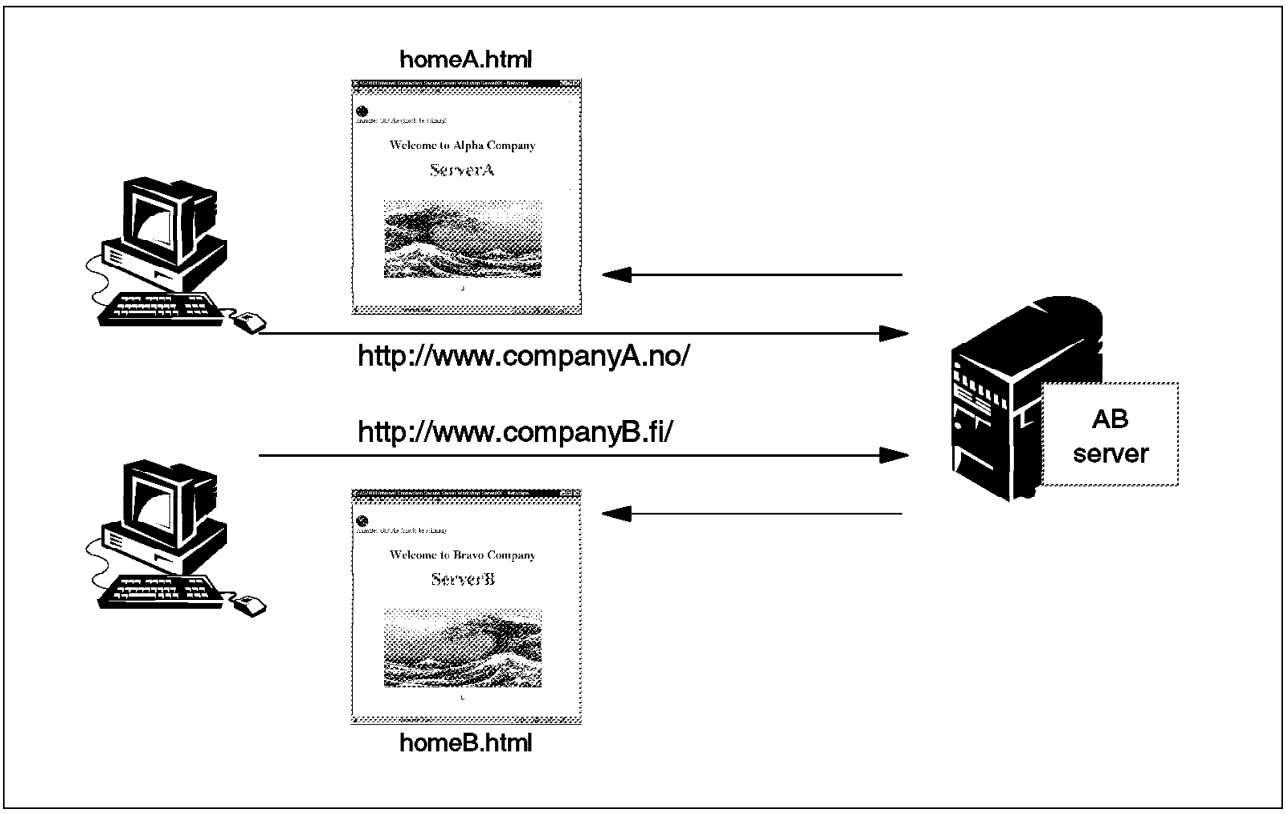


Figure 30. Running the Server with Multiple IP Addresses

The way you configure the server to serve different information for each customer is by indicating that certain parts of the configuration apply only to requests coming in on certain addresses (or addresses that match a template). You can configure three key parts of the server such that requests are processed differently based on the IP address they come in on:

- Welcome pages:

  You can specify a different set of file names to use as welcome pages depending on the address a request comes in on. The file names you define as welcome pages determine how the server responds to requests that do not contain a file name.

  For example, you might want to specify that homeA.html is a welcome page only for requests received on 9.67.106.79; and homeB.html is a welcome page only for requests received on an address matching the template 9.83.*.

  From the Configuration and Administration forms page, you can configure the list of welcome pages by clicking on **Initial Page**. From the Initial Page form, click the help icon for information on defining welcome pages and how to associate a welcome page file name with an IP address or IP address template.

- Mapping rules:

  You can specify a different set of mapping rules for the server to use depending on the address a request comes in on. Mapping rules map a request to a physical file on the server and determine whether the server processes a request.

  For example, you might want to specify that a request beginning with /cgi-bin/ received on address 9.67.106.79 is mapped to the /companyA/cgi/

directory; and the same request received on an address matching the template 9.83.* is mapped to the /companyB/cgi/ directory.

From the Configuration and Administration forms page, you can configure the mapping rules by clicking on **Request Routing**. From the Request Routing form, click on the help icon for information on how to use mapping rules and how to associate a mapping rule with an IP address or IP address template.

- Access Control:

  You can activate different protection rules for a request based on the address the request comes in on. Protection rules are defined in protection setups and determine how the server controls access to files and programs. For more information, see Chapter 8, “Protecting Server Resources” on page 121.

6.1.2 Some Things You Should Know Before You Start

How each server instance is configured is determined by two sets of information:

- Instance information
- Configuration file

Each server instance is defined by instance information, which contains information specific to that server instance. In addition, the configuration file contains values for configuration directives, which affect any server instance that uses that configuration file.

You can define startup values for server instances in several different places. The following table should clarify the precedence order of these values (see Table 3).

Table 3. Precedence Order for Configuration Information

Configuration Information (1=Highest Precedence; 5=Lowest)	**How to Specify Values from the Browser**	**How to Specify Values using AS/400 Commands**
1. Instance start-up values	Cannot specify here	HTTPSVR parameter on STRTCPSVR command
2. Instance parameters	Server Instance form	Cannot specify here
3. Configuration file [1]	Configuration and Administration form	WRKHTTPCFG command
4. Global attribute values	General Configuration and Administration form	CHGHTTPA command
5. Server defaults	Cannot specify here	Cannot specify here
[1] The configuration file is named in the instance parameters.		

6.2 Basic Configuration

In this section, we use the ADMIN server configuration interface to perform basic configuration. See Chapter 5, “Web Browser Configuration Interface” on page 33 for information on the ADMIN server configuration interface.

6.2.1 General Configuration and Administration

The general configuration and administration form is used to specify which server configuration to change, to create a new server configuration, or to go to the form for changing the global attribute values.

Multiple HTTP servers may be defined and running concurrently on an AS/400 system. Each of those servers is called an instance and is named. This form lets you select an instance and the action to perform on that instance. Once you select an instance, you can change the configuration, delete the instance, start, stop, or restart that instance. In addition, instances can be created using this form.

Global attribute values define default server characteristics on a system-wide level. This form allows you to go to the form for changing those values.

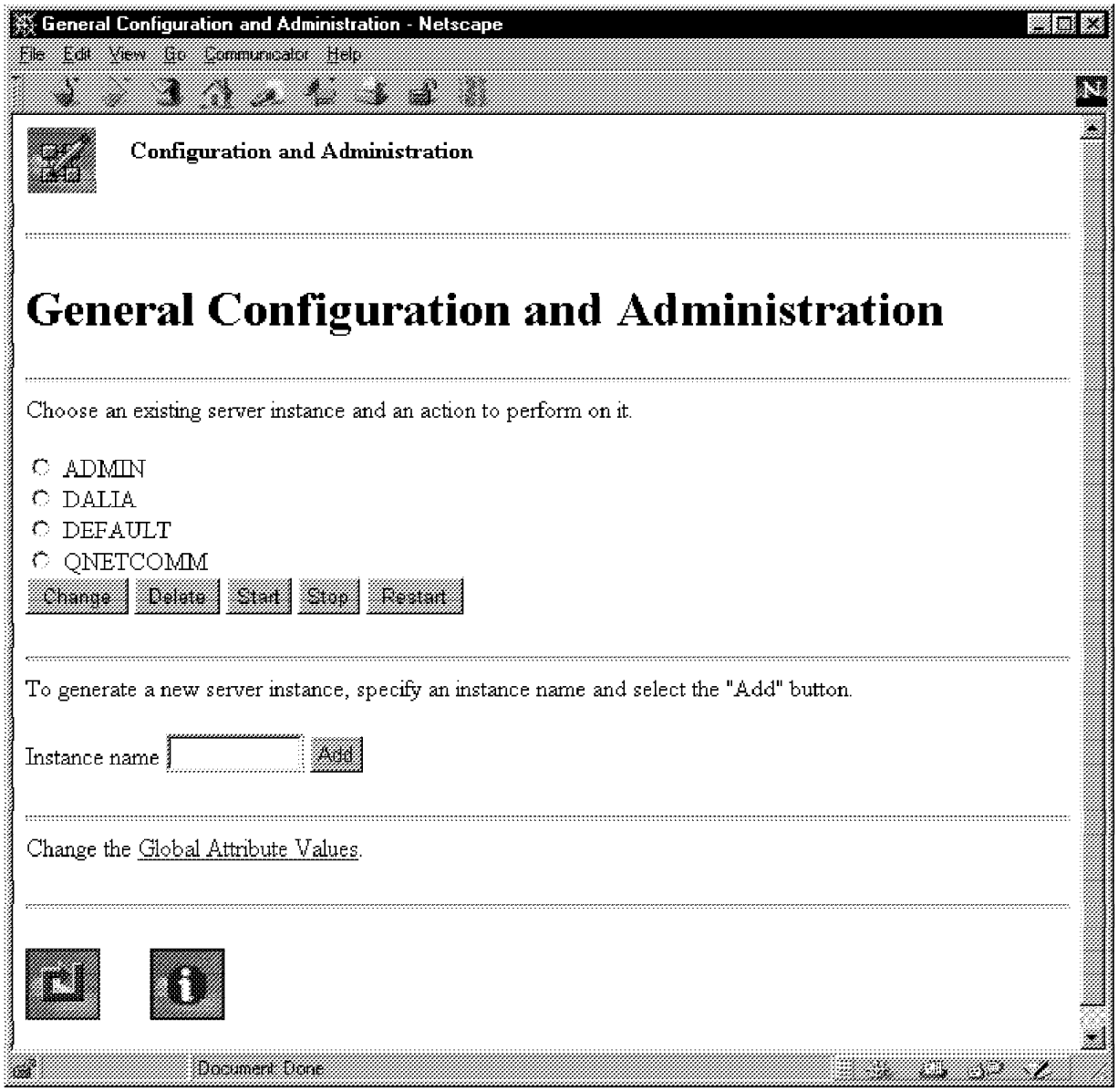


Figure 31. General Configuration and Administration

To modify an instance:

1. Select the server instance you want to modify.
2. Click **Change** and the Server Instance form is shown (see Figure 34 on page 54).

To delete an instance:

1. Select the server instance you want to delete.
2. Click **Delete**.

Important!

Be careful as *no* confirmation is requested prior to the instance being deleted.

To start an instance:

1. Select the server instance you want to start.
2. Click **Start**.

To stop an instance:

1. Select the server instance you want to stop.
2. Click **Stop**.

To restart an instance:

1. Select the server instance you want to restart.
2. Click **Restart**.

Note: If the Confirmation page tells you to restart the server to invoke the configuration changes you just made, you can return to the General Configuration and Administration page by pressing the **General** button on the Confirmation page and return to this page to restart the server. But if the Confirmation page does not tell you this, then you need to stop the server and start it again for the configuration changes to take effect.

To add an instance:

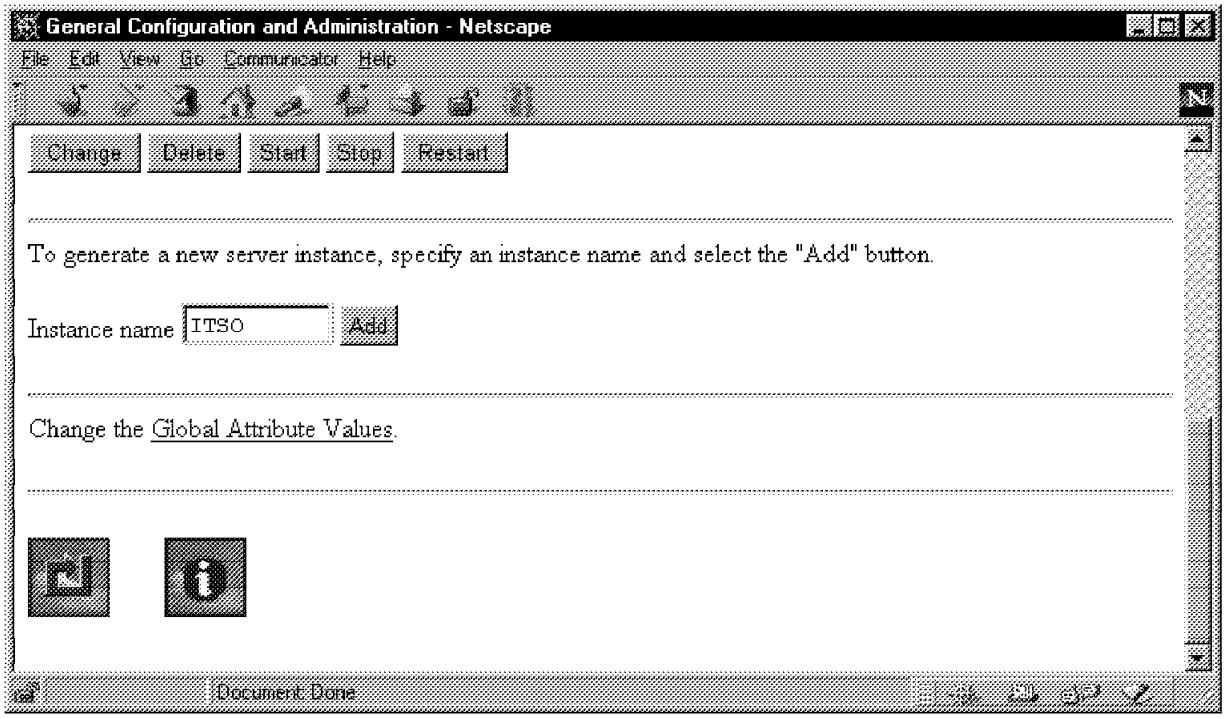


Figure 32. Creating a New Server Instance

1. Enter the new instance name in **Instance name**. The name must be an AS/400 member name ranging from 1-10 characters.
2. Click **Add** and the Server Instance form is shown (see Figure 34 on page 54).

Note: Adding an instance results in a member being added to QUSRSYS/QATMHINSTC.

To modify the global attribute values to be used by all server instances:

1. Click **Global Attribute Values** and the Global Attribute Values form is shown (see Figure 33 on page 52).

6.2.2 Global Attribute Values

Use this form to change the global attribute values to be used by all server instances. The attributes may be overridden for a particular instance.

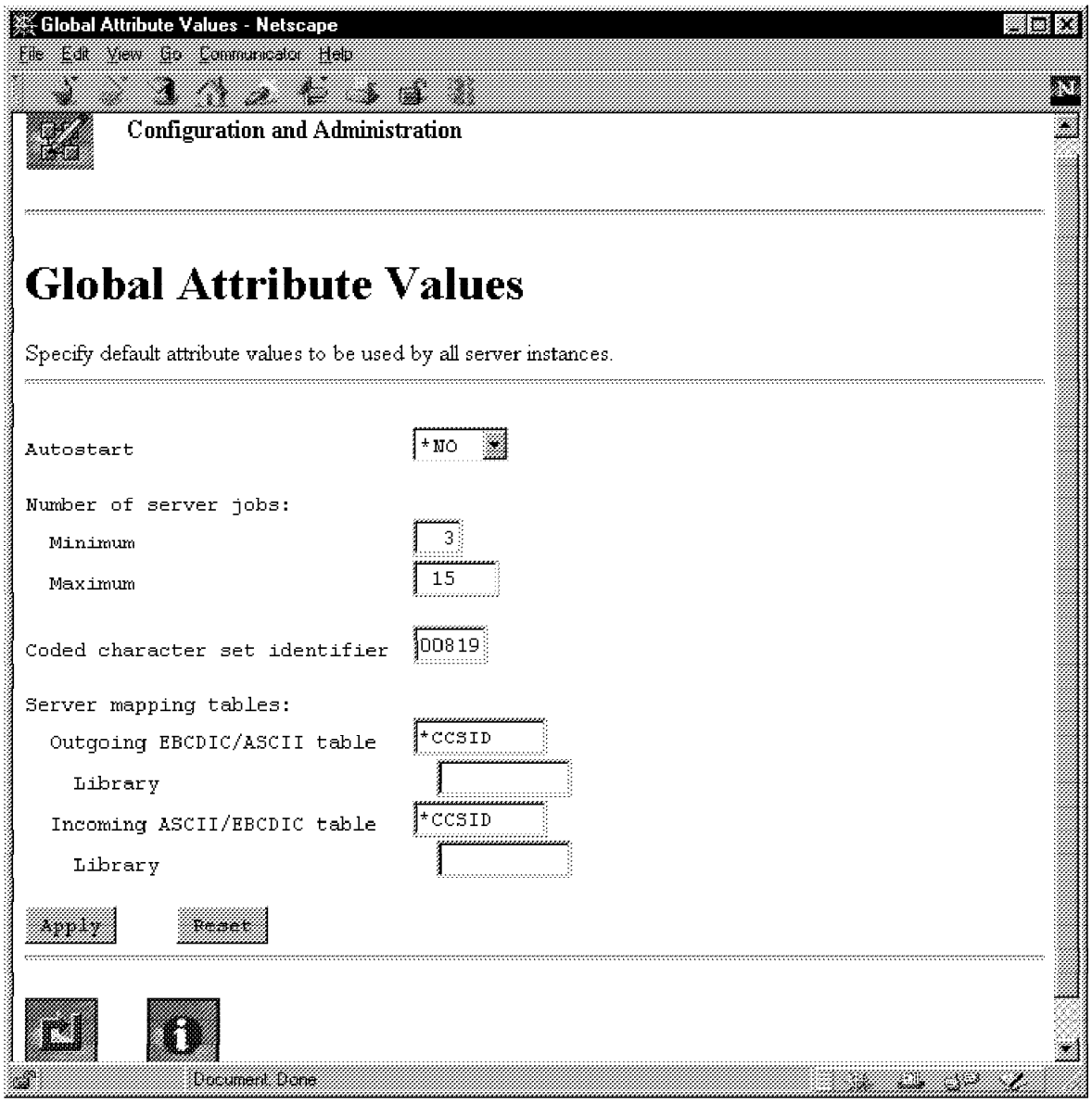


Figure 33. Global Attribute Values

1. Fill in the following fields as required:
   - **Autostart** - indicate whether you want this server instance to be automatically started:
     - YES - specifies that you want the server to start whenever TCP/IP is started. If you use the STRTCPSVR command, this parameter is ignored and the server is started.
     - NO - specifies that you do not want this server instance started when TCP/IP is started.
   - **Number of server jobs** (see Figure 68 on page 99):

- **Minimum** - specify the default minimum number of server jobs to start when the server instance is started and the minimum number the server is to keep open. It can be overridden for a particular server instance through instance start-up values, instance parameters, or a configuration file directive.
- **Maximum** - specify the default maximum number of server jobs the server instance is to keep open. If you set the maximum number of server jobs too high, the performance of the server instance in satisfying browser requests is affected. However, if the server instance is serving large documents or programs, you might need to increase the value specified for the maximum number of server jobs. It can be overridden for a particular server instance through instance start-up values, instance parameters, or a configuration file directive.
  - *NOMAX - There is no limit to the maximum number of server jobs allowed to start.

- **Coded character set identifier** - indicate the ASCII coded character set identifier (CCSID) that the server uses by default to translate from ASCII to EBCDIC and from EBCDIC to ASCII. The server instance uses this value when a character set and code page are not identified in the MIME header from the Web browser. It can be overridden for a particular server instance through instance start-up values, instance parameters, or a configuration file directive.
  - *DFT option can be used to specify a CCSID value of 00819 (ISO 8859-1 8-bit ASCII).
- **Server mapping tables:**
  - **Outgoing EBCDIC/ASCII table** - specify the table that you want the server to use for character conversion for the outgoing EBCDIC to ASCII tables. It can be overridden for a particular server instance through instance parameters or configuration file directive.
    - *CCSID - specifies that the *CCSID parameter is used to determine outgoing mapping.
  - **Library** - library name is required when specifying an outgoing table.
  - **Incoming EBCDIC/ASCII table** - specify the table that you want the server to use for character conversion for the incoming ASCII to EBCDIC tables. It can be overridden for a particular server instance through instance parameters or configuration file directive.
    - *CCSID - specifies that the *CCSID parameter is used to determine incoming mapping.
  - **Library** - library name is required when specifying an incoming table.

2. Click **Apply** to update the server with the changes you made to the form, or click **Reset** to return to the values that were on the form before you made the changes.

6.2.3 Server Instance

After you click the General Configuration and Administration form **Change** button, the Server Instance form is shown.

Use this form to:

1. Define which configuration is associated with a server instance.
2. Create a new configuration to be associated with a server instance.
3. Modify a specific configuration, or update the instance parameters for a given server instance.

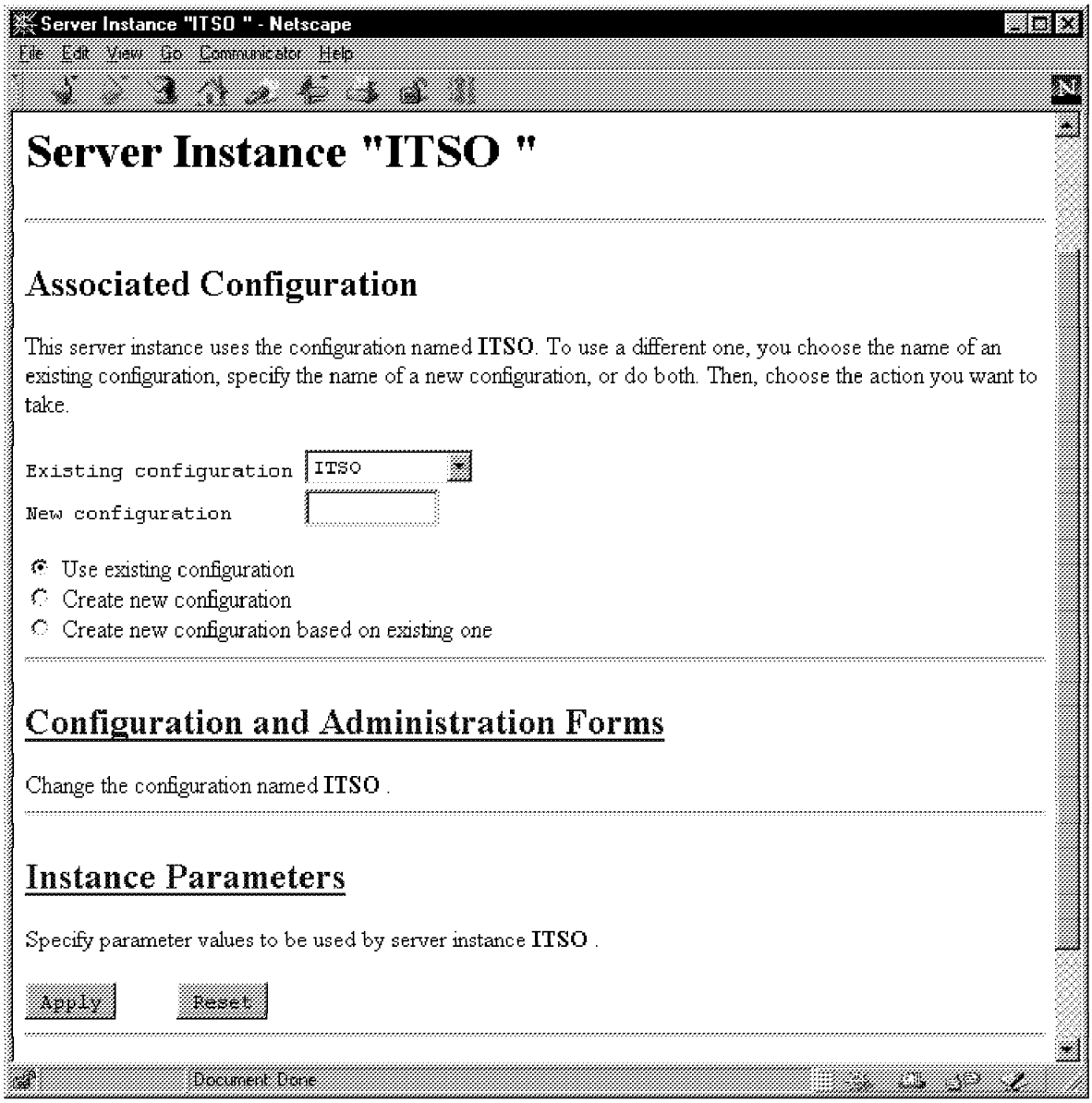


Figure 34. Server Instance

To modify an existing configuration:

1. Select the configuration you want to modify.
2. Select **Use existing configuration**.

3. Click **Configuration and Administration Forms**.

To create a new configuration:

1. Enter the new configuration name in the **New configuration** field. The name must be an AS/400 member name ranging from 1-10 characters.
2. Select **Create new configuration**.
3. Click **Apply** to update the server with the changes you made to the form, or click **Reset** to return to the values that were on the form before you made the changes.

Note: Adding a new configuration results in a member being added to QUSRSYS/QATMHTTPC.

To create a new configuration based upon an existing configuration:

1. Select the existing configuration.
2. Enter the new instance name in the **New configuration** field. The name must be an AS/400 member name, ranging from 1-10 characters.
3. Select **Create new configuration based on existing one**.
4. Click **Apply** to update the server with the changes you made to the form, or click **Reset** to return to the values that were on the form before you made the changes.

To change the instance parameters for a selected server instance:

1. Select **Instance Parameters**.
2. Fill in the **Instance Parameters** form (see Figure 35 on page 56).

Note

Multiple server instances can share a single configuration file or each can have a unique configuration file. *Associated Configuration* associates a configuration to a server instance.

The instance parameters take precedence over parameters in the server configuration file. These parameters also take precedence over the global attribute values.

As an example, if you have a configuration you want to share among several servers, but want to change the ports associated with those servers, you can use this form to specify a different port number for a particular instance.

6.2.4 Instance Parameters

Use this form to specify the parameters that apply only to a particular server instance.

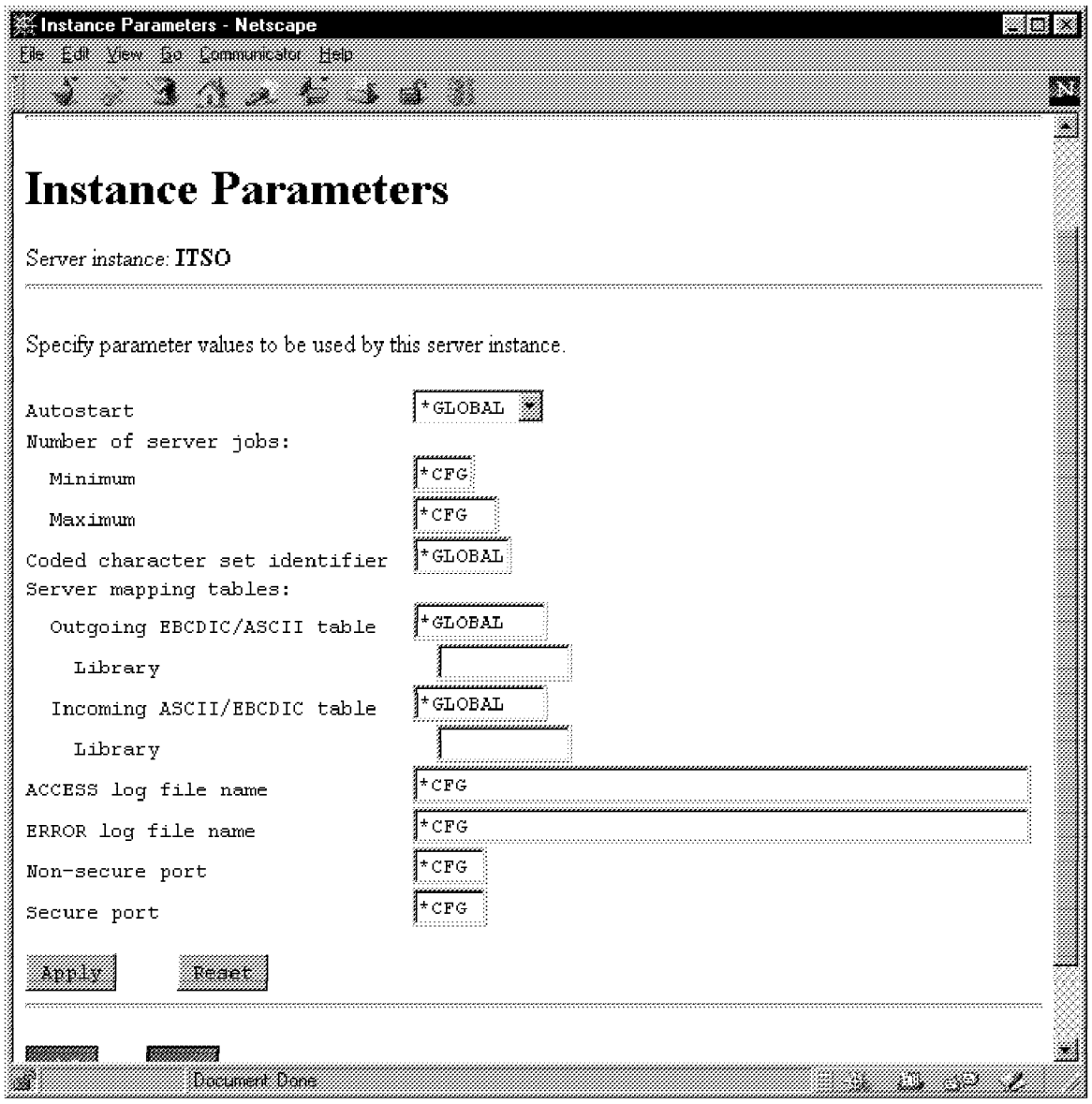


Figure 35. Instance Parameters

To specify parameters to be used by this server instance:

1. Fill in any of the following optional fields:
   - **Autostart** - indicate whether you want this server instance to be automatically started.
     - YES - specifies that you want the server to start whenever TCP/IP is started. If you use the STRTCPSVR command, this parameter is ignored and the server is started.
     - NO - specifies that you do not want this server instance started when TCP/IP is started.

  - *GLOBAL - specifies that you want this server instance to use whatever **Autostart** value is defined in the Global Attributes form (see Figure 33 on page 52).

- **Number of server jobs** (see Figure 68 on page 99):
  - **Minimum** - specify the minimum number of server jobs to start when the server instance is started and the minimum number the server is to keep open. It can be overridden through instance start-up values (see 6.11, "Start TCP/IP Server and End TCP/IP Server Commands" on page 102).
    - *CFG - specifies that you want this server instance to use the Minimum value set in the Jobs form (see Section 6.10.1, "Jobs" on page 99).
  - **Maximum** - specify the maximum number of server jobs the server instance is to keep open. If you set the maximum number of server jobs too high, the performance of the server instance in satisfying browser requests is affected. However, if the server instance is serving large documents or programs, you might need to increase the value specified for the maximum number of server jobs. It can be overridden through instance start-up values (see 6.11, "Start TCP/IP Server and End TCP/IP Server Commands" on page 102).
    - *NOMAX - There is no limit to the maximum number of server jobs allowed to start.
    - *CFG - specifies that you want this server instance to use the maximum value set in the Jobs form (see Section 6.10.1, "Jobs" on page 99).
- **Coded character set identifier** - indicate the ASCII coded character set identifier (CCSID) that the server uses to translate from ASCII to EBCDIC and from EBCDIC to ASCII. The server instance uses this value when a character set and code page are not identified in the MIME header from the Web browser. It can be overridden through instance start-up values (see 6.11, "Start TCP/IP Server and End TCP/IP Server Commands" on page 102).
  - *GLOBAL - specifies that you want this server instance to use a coded character set identifier value defined in the Global Attributes form (see Figure 33 on page 52).
- **Server mapping tables:**
  - **Outgoing EBCDIC/ASCII table** - specify the table that you want the server to use for character conversion for the outgoing EBCDIC to ASCII tables.
    - *CCSID - specifies that the *CCSID parameter is used to determine outgoing mapping.
    - *GLOBAL - specifies that you want this server instance to use the Outgoing EBCDIC/ASCII table value defined in the Global Attributes form (see Figure 33 on page 52).
  - **Library** - library name is required when specifying an outgoing table. It is not a requirement for *GLOBAL or *CCSID options.
  - **Incoming EBCDIC/ASCII table** - specify the table that you want the server to use for character conversion for the incoming ASCII to EBCDIC tables.

- *CCSID - specifies that the *CCSID parameter is used to determine incoming mapping.
- *GLOBAL - specifies that you want this server instance to use the Incoming EBCDIC/ASCII table value defined in the Global Attributes form (see Figure 33 on page 52).

– **Library** - library name is required when specifying an incoming table. It is not a requirement for *GLOBAL or *CCSID options.

- **ACCESS log file name** - specify the name of the file where the access log is to be stored. It can be overridden through instance start-up values (see 6.11, “Start TCP/IP Server and End TCP/IP Server Commands” on page 102).
  – *CFG - specifies that you want this server instance to use the access log file name set in the Access Log File Configuration form (see Figure 52 on page 74).
- **ERROR log file name** - specify the name of the file where the error log is to be stored. It can be overridden through instance start-up values (see 6.11, “Start TCP/IP Server and End TCP/IP Server Commands” on page 102).
  – *CFG - specifies that you want this server instance to use the error log file name set in the Error Log File Configuration form (see Figure 55 on page 78).
- **Non-secure port** - specify the port number the server instance should listen to for requests for non-secure files. The standard port number is 80. Other port numbers less than 1024 are reserved for other TCP/IP applications and should not be used. When a port other than 80 is used, clients are required to include a specific port number on requests to the server. It can be overridden through instance start-up values (see 6.11, “Start TCP/IP Server and End TCP/IP Server Commands” on page 102).
  – *CFG - specifies that you want this server instance to use the Non-secure port set in the Basic Configuration form (see Figure 38 on page 61).
- **Secure port** - specify the port number the server instance should listen to for requests for secure files. It can be overridden through instance start-up values (see 6.11, “Start TCP/IP Server and End TCP/IP Server Commands” on page 102).
  – *CFG - specifies that you want this server instance to use the Secure port set in the Security Configuration form (see Figure 133 on page 161).

2. Click **Apply** to update the server with the changes you made to the form, or click **Reset** to return to the values that were on the form before you made the changes.

6.2.5 Configuration and Administration Forms

The Configuration and Administration Forms page is the main menu for configuring a server instance. Through this menu, you have access to the configuration pages that are introduced in the following sections and chapters. Having selected the server configuration from the Server Instance form (see Figure 34 on page 54), click on **Configuration and Administration Forms**.

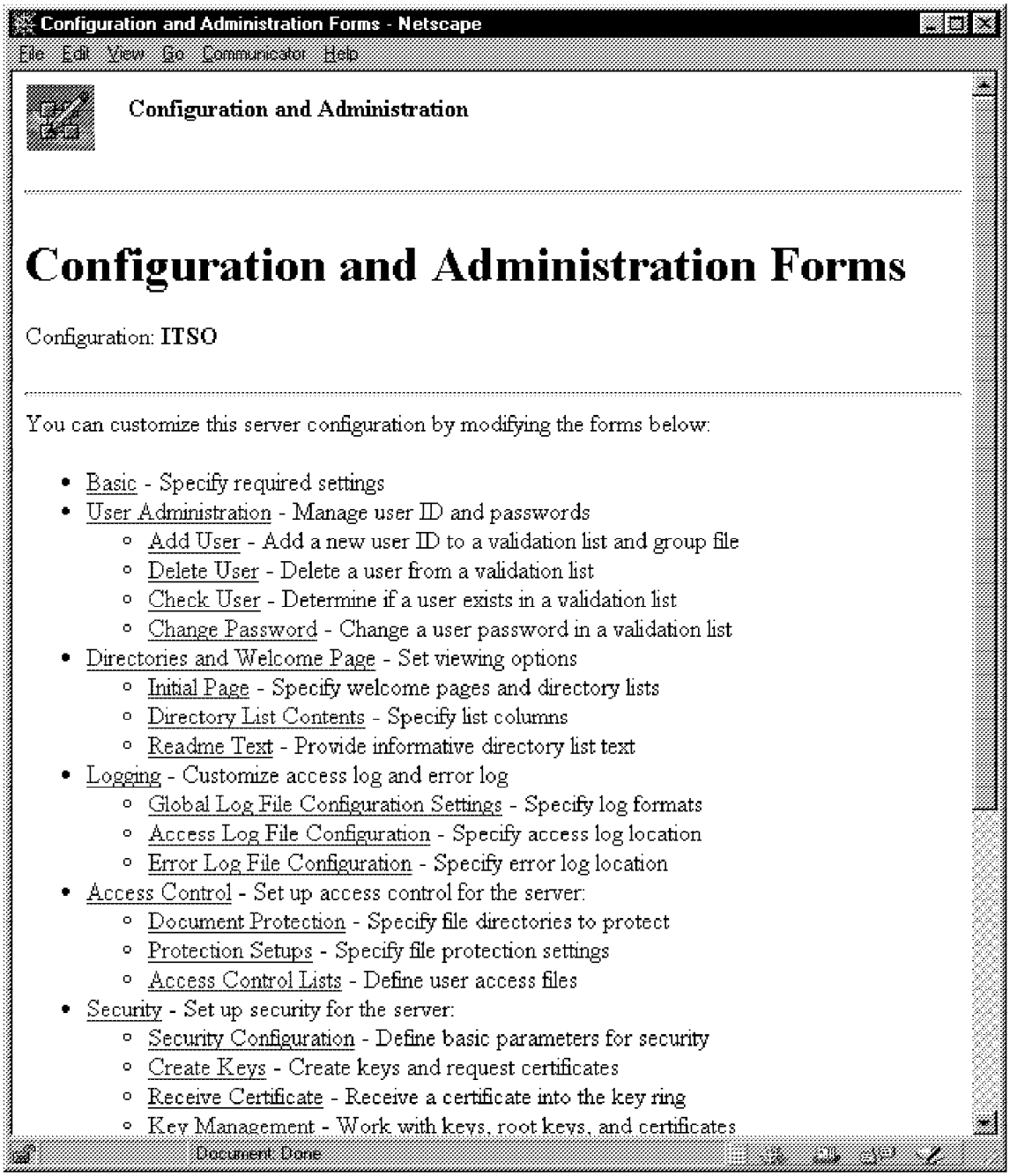


Figure 36. Configuration and Administration Forms - Part 1

Note: Security options (Security, Security Configuration, Create Keys, Receive Certificate, and Key Management) are only shown when ICSS for AS/400 is installed.

The configuration pages under User Administration and Access Control are covered in Chapter 8, “Protecting Server Resources” on page 121. The configuration pages under Security are covered in Chapter 9, “Establishing a Secure Connection” on page 159.

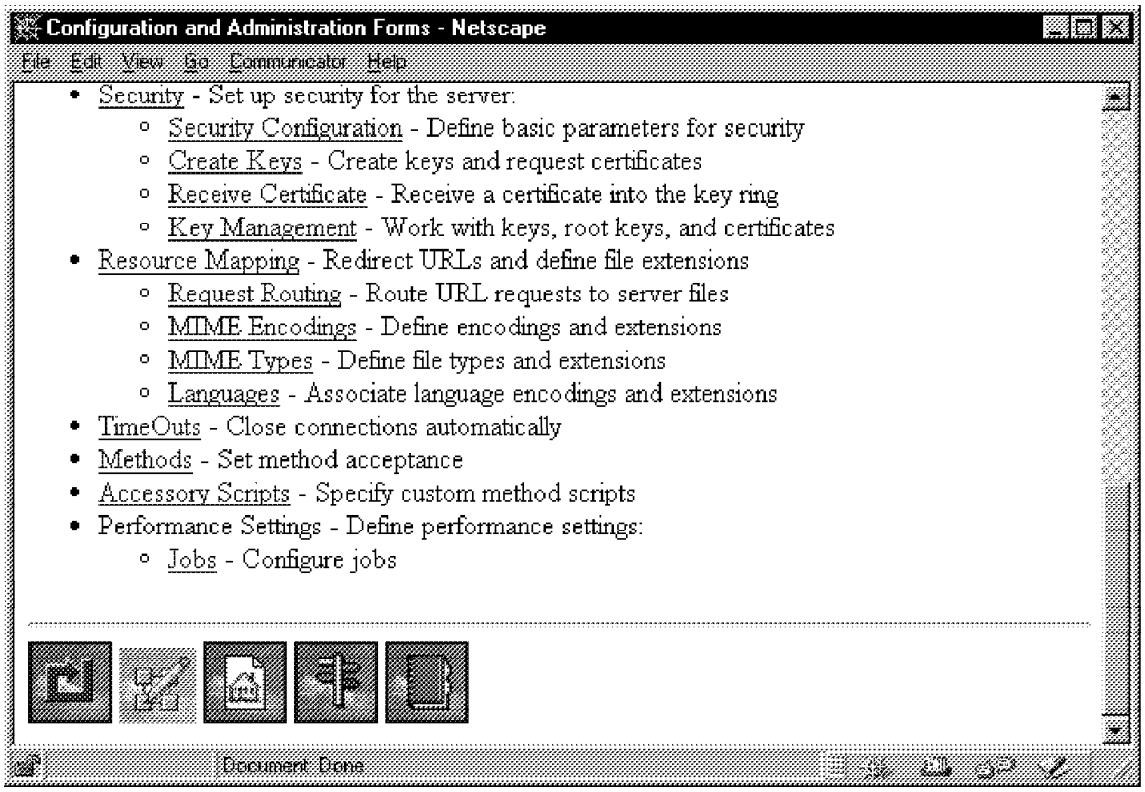


Figure 37. Configuration and Administration Forms - Part 2

6.3 Basic Settings

Use this form to change the server basic settings.

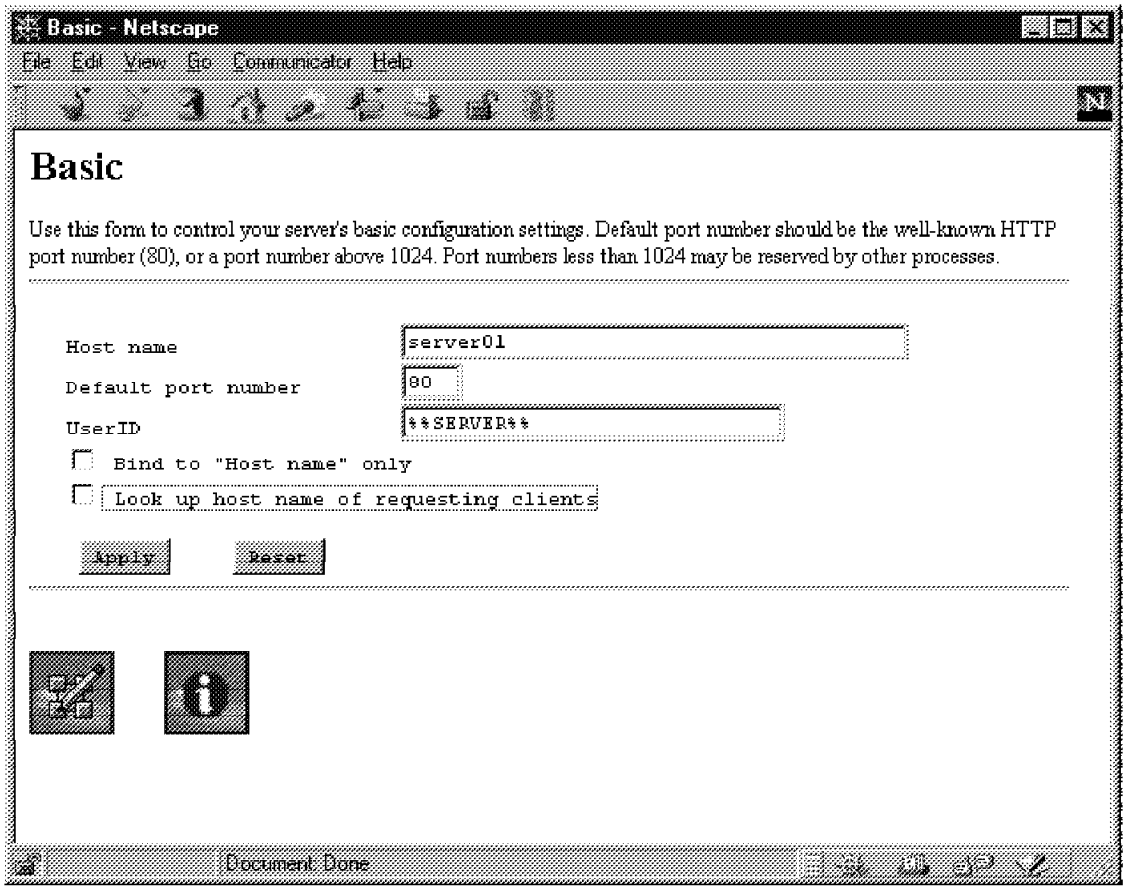


Figure 38. Basic Settings

To change the basic settings:

1. Fill in the following fields:
   - **Host name** - specify the host name or an IP address to be returned to clients from document requests. If you use **Bind to ″Host name″ only** to bind to a specific address, this host has to be a valid host name that can be resolved from a domain name server.
   - **Default port number** - specify the port number you want the server to listen to. The default port number can be overridden through instance parameters (see Figure 35 on page 56) or by the -p option in the STRTCPSVR command (see 6.11, “Start TCP/IP Server and End TCP/IP Server Commands” on page 102). The valid range of port numbers is 1 to 65535. The well known port number for Hypertext Transfer Protocol (HTTP) is 80. Other port numbers less than 1024 are reserved for other TCP/IP applications and should not be used. Port numbers 8080 and 8008 are commonly used for testing servers.

Reminder

Before you define a port other than 80 for the server, please check the availability of the port from *Assigned Numbers*, RFC1700 (or newer), (RFCs are available from http://ds.internic.net/).

- **UserID** - use UserID to specify an AS/400 user profile that the server switches to while completing the HTTP transaction. Specify %%SYSTEM%% to use the profile of the server, which is QTMHHTTP by default. Specify %%CLIENT%% to use the user profile supplied by the client; the client is prompted for a user ID and password. This user profile is used to retrieve objects.
- **Bind to ″Host name″ only** - check this box to run a different server instance on each IP address or host name on a multi-networking system. All the servers may listen on the same port. If you check this box, the server binds to the IP address specified in the HostName directive only instead of binding to all local IP addresses.
- **Look up host name of requesting clients** - check this box if you want the server to look up the host name of requesting clients. This option allows you to use host names on address templates in protection setups, but reduces the performance of the server slightly. If you do not check this box, the server identifies clients by their IP address instead of by their host name and uses the IP address in template protections.

2. Click **Apply** to update the server with the changes you made to the form, or click **Reset** to return to the values that were on the form before you made the changes.

UserID Object Authority

The following authorities apply if you define %%SERVER%% as **UserID** in the previous form:

- The QTMHHTTP user profile or *PUBLIC must be granted *USE authority to all AS/400 library system objects that you intend to serve.
- The QTMHHTTP user profile or *PUBLIC must be granted *RX authority to all QDLS and integrated file system objects that you intend to serve.
- To access any object using a CGI program, the QTMHHTP1 user profile or *PUBLIC needs the same authority to the objects as QTMHHTTP.

If you do not define %%SERVER%% as **UserID**, the preceding authorities apply to the userid that is defined.

```
              HTTP Configuration File

         HostName       server01
         Port       80
         UserID       %%SERVER%%
         BindSpecific       Off
         DNS-Lookup       Off
```

Figure 39. Basic Settings Added to HTTP Configuration File

Figure 39 shows the results of applying the settings in the form shown in Figure 38 on page 61.

6.4 Directory and Welcome Page

Figure 40. Directory and Welcome Page

Through this form, we define:

- Initial page
- Directory list viewing
- Directory list contents
- Readme text

6.4.1 Initial Page

Use this form to:

- Create or modify a list of file names that you want the server to recognize as welcome pages.
- Define which directories you want clients to be able to browse.

The server returns a welcome page for client requests that do not include a specific file name. If the request contains a directory name, the server searches that directory for a welcome page. If the request does not contain a directory name and no PASS directives are configured to point to a directory where the welcome file is stored, directory list processing is started. See the Figure 44 on page 68 for details on welcome file and directory list processing. The order of the list is important because the server might find more than one welcome page in the directory it is searching. The first listed welcome page (a match) is the one the server returns.

You can specify a different set of file names to use as welcome pages depending on the address a request comes in on. The file names you define as welcome pages determine how the server responds to requests that do not contain a file name.

For example, you might want to specify that `homeA.html` is a welcome page only for requests received on `9.67.106.79`; and `homeB.html` is a welcome page only for requests received on an address matching the template `9.83.*`.

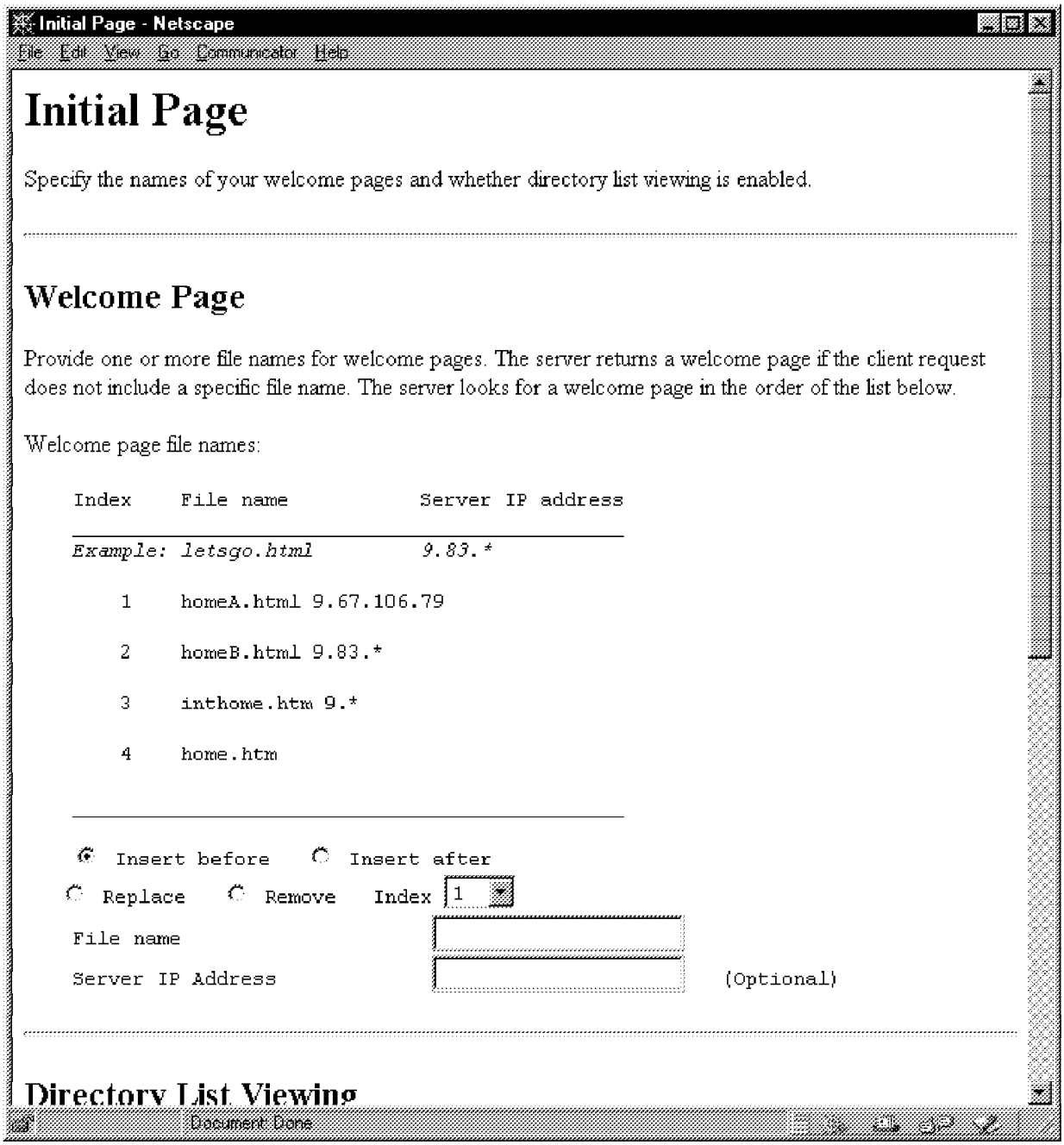


Figure 41. Initial Page - Welcome Page

To insert a new welcome page:

1. Select either **Insert before** or **Insert after**.

2. Select an **Index number**:
    - Your choices in step 1 and step 2 indicate the position you want the item to have in the list. For example, if you select **Insert before** and **Index 2**, the item is second in the list. If you select **Insert after** and **Index 4**, the item is fifth in the list.
3. Enter the new welcome page file name in the **File name** field.
4. Optionally specify the server IP address to associate with the welcome page.
5. Click **Apply** to update the server with the changes you made to the form, or click **Reset** to return to the values that were on the form before you made the changes.

To replace a welcome page:

1. Select **Replace**.
2. Select the **Index number** of the item you want to replace.
3. Enter the file name of a new welcome page in the **File name** field.
4. Optionally specify the server IP address to associate with the welcome page.
5. Click **Apply** to update the server with the changes you made to the form, or click **Reset** to return to the values that were on the form before you made the changes.

To remove a welcome page:

1. Select **Remove**.
2. Select the **Index number** of the item you want to remove.
3. Click **Apply** to update the server with the changes you made to the form, or click **Reset** to return to the values that were on the form before you made the changes.

6.4.2 Directory List Viewing

You can specify whether you want clients to be able to view listings of the server′s directories. You can either allow viewing of all directories, or allow viewing of only specific directories. The server responds to requests as follows:

- For requests made to directories that do not allow directory list viewing, the server always looks for a welcome page to return.
- For requests made to directories that do allow directory list viewing, the server first checks the last character of the request′ s URL. If the last character is a slash (/), the server looks for a welcome page to return. If the last character is not a slash, the server returns a listing of the directory.
- For requests that contain only the server name, the server always looks for a welcome file in the directory that you pass these requests to. The server does not generate a listing for that directory.

Scroll down for Directory List Viewing.

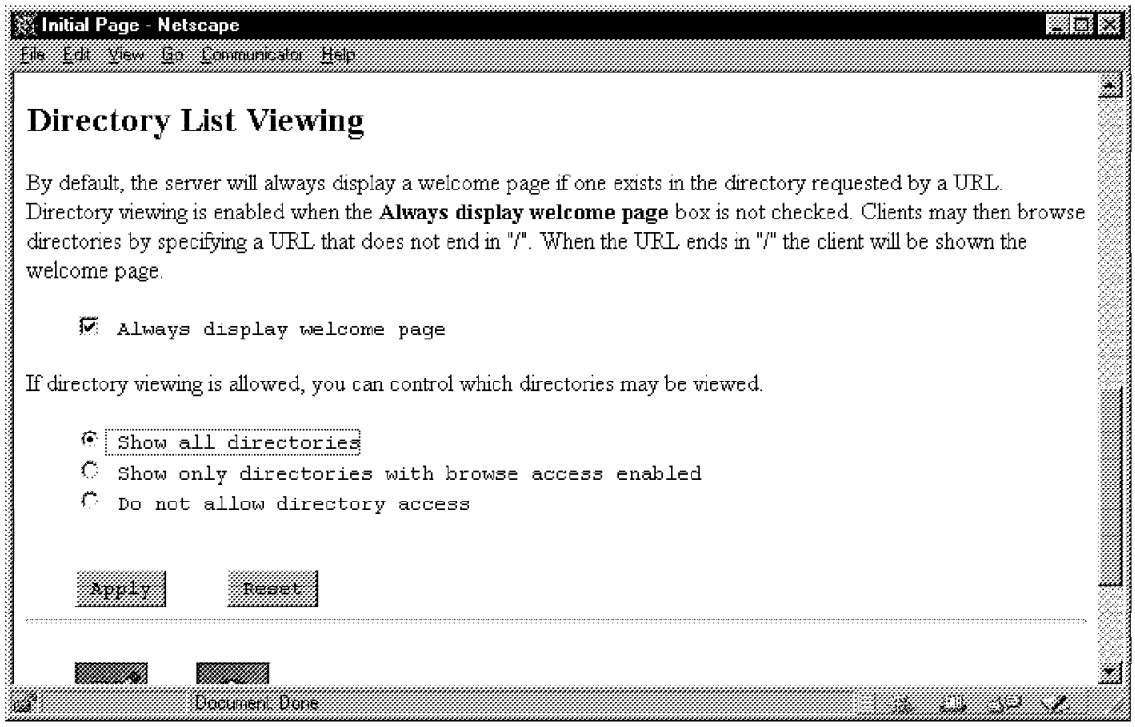


Figure 42. Initial Page - Directory List Viewing

To allow viewing for all directories:

1. Uncheck **Always display welcome page**.
2. Select **Show all directories**.
3. Click **Apply** to update the server with the changes you made to the form, or click **Reset** to return to the values that were on the form before you made the changes.

To allow viewing of only certain directories:

1. Uncheck **Always display welcome page**.
2. Select **Show only directories with browse access enabled**.
3. Click **Apply** to update the server with the changes you made to the form, or click **Reset** to return to the values that were on the form before you made the changes.

Note: You need to have a file or a file member called `wwwbrws` in the target directory or file to enable the viewing.

To not allow viewing of any directories:

1. Select **Do not allow directory access**.
2. Click **Apply** to update the server with the changes you made to the form, or click **Reset** to return to the values that were on the form before you made the changes.

```
               HTTP Configuration File

          Welcome    homeA.html 9.67.106.79
          Welcome    homeB.html 9.83.*
          Welcome    inthome.htm 9.*
          Welcome    home.htm
          AlwaysWelcome       On
          DirAccess       On
```

Figure 43. Initial Page Settings Added to HTTP Configuration File

Figure 43 shows the results of applying the settings in the form shown in Figure 41 on page 64 and Figure 42 on page 66.

The following figure explains the processing of welcome page and directory listing directives:

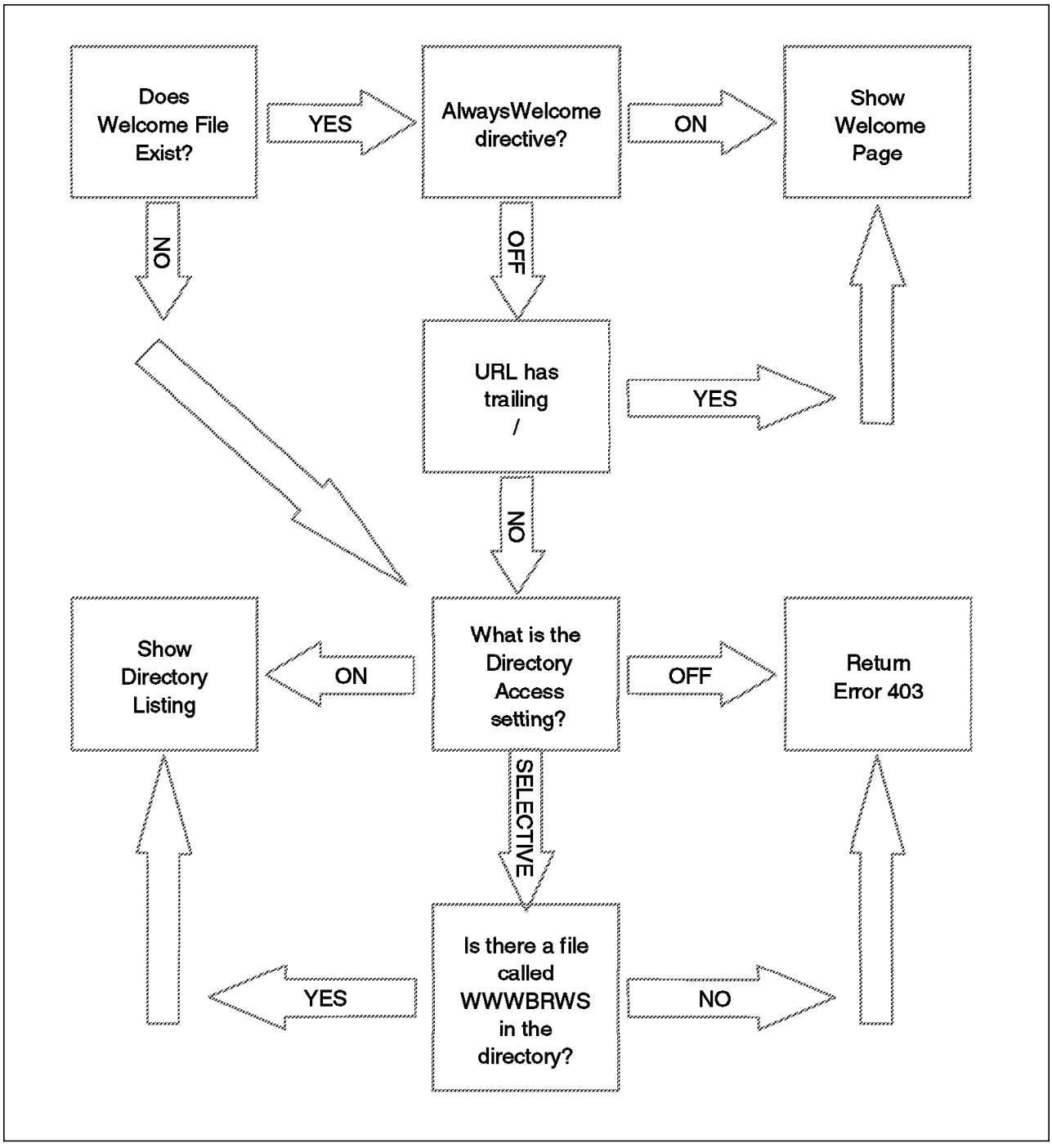


Figure 44. Welcome File and Directory Processing

6.4.3 Directory List Contents

Use this form to control the directory list contents the server displays to clients such as:

- Files
- Fields
- Order of files

A directory listing shows the directory's files and sub-directories. Each file or subdirectory item is shown on a separate line.

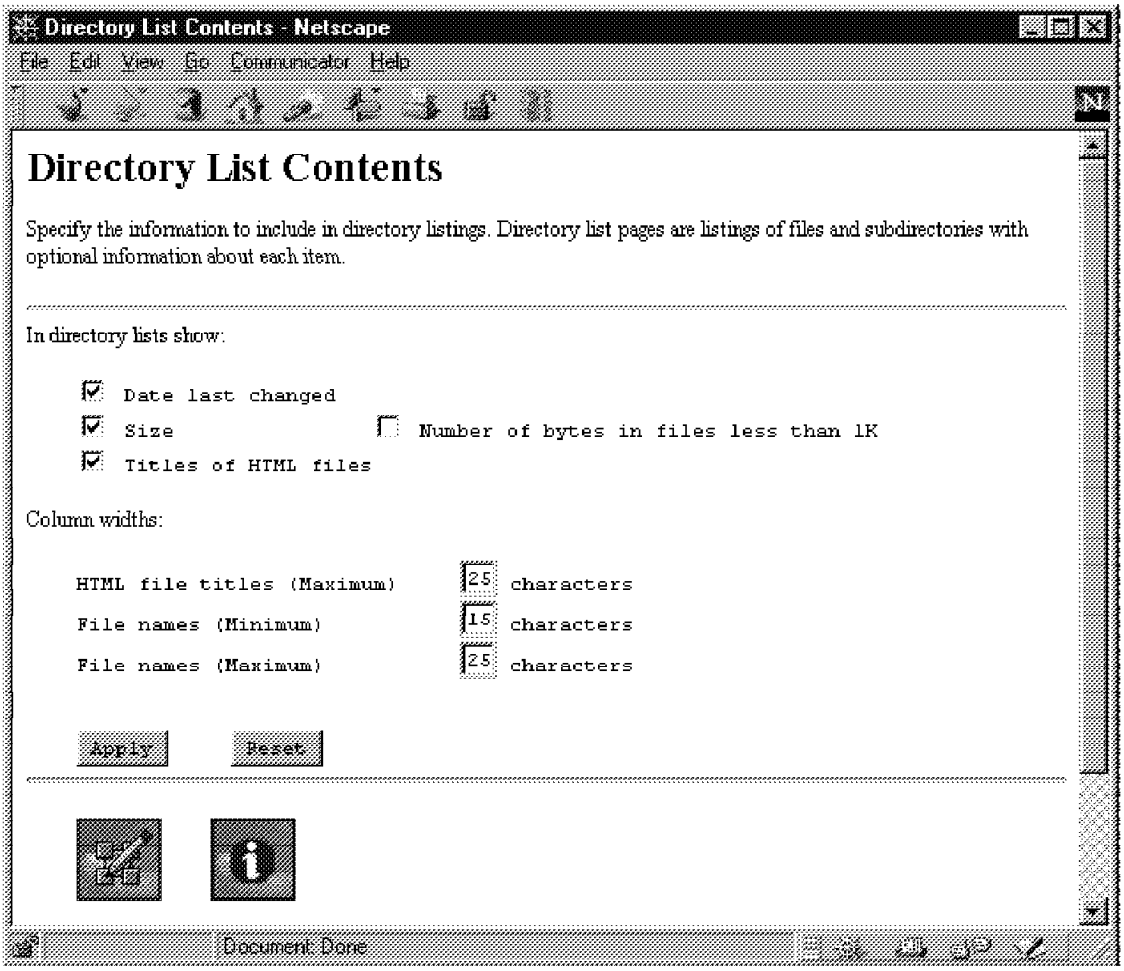


Figure 45. Directory List Contents

To define the contents of the directory lists:

1. Select as many as you want to use:
   - **Date last changed** - show the last date each file was changed.
   - **Size** - show the size of each file.
   - **Number of bytes in files less than 1K** - show the exact number of bytes for files smaller than 1K (otherwise, these files are shown with a size of 1K).
   - **Titles of HTML files** - show the titles of HTML files (titles are taken from the text between the HTML `<title>` and `</title>` tags).

To define title and name column widths: You can control how much space you want directory listings to use for file titles and names.

1. Fill in the following fields:
   - **HTML file titles (Maximum)** - the maximum number of characters to use for displaying titles of HTML files.
   - **File names (Minimum)** - the minimum number of characters to use for displaying file names.
   - **File names (Maximum)** - the maximum number of characters to use for displaying file names.

Click **Apply** to update the server with the changes you made to the form, or click **Reset** to return to the values that were on the form before you made the changes.

```
               HTTP Configuration File

          DirShowDate       On
          DirShowSize       On
          DirShowBytes       Off
          DirShowDescription       On
          DirShowMaxDescrLength       25
          DirShowMinLength       15
          DirShowMaxLength       25
```

Figure 46. Directory List Contents Added to HTTP Configuration File

Figure 46 shows the results of applying the settings in the form shown in Figure 45 on page 69.

6.4.4 Readme Text

Use this form to control the display of README information on the server's directory listings. When the server creates a directory listing, it looks in the directory for a file named README. The README file typically contains a brief description of the contents of the directory.

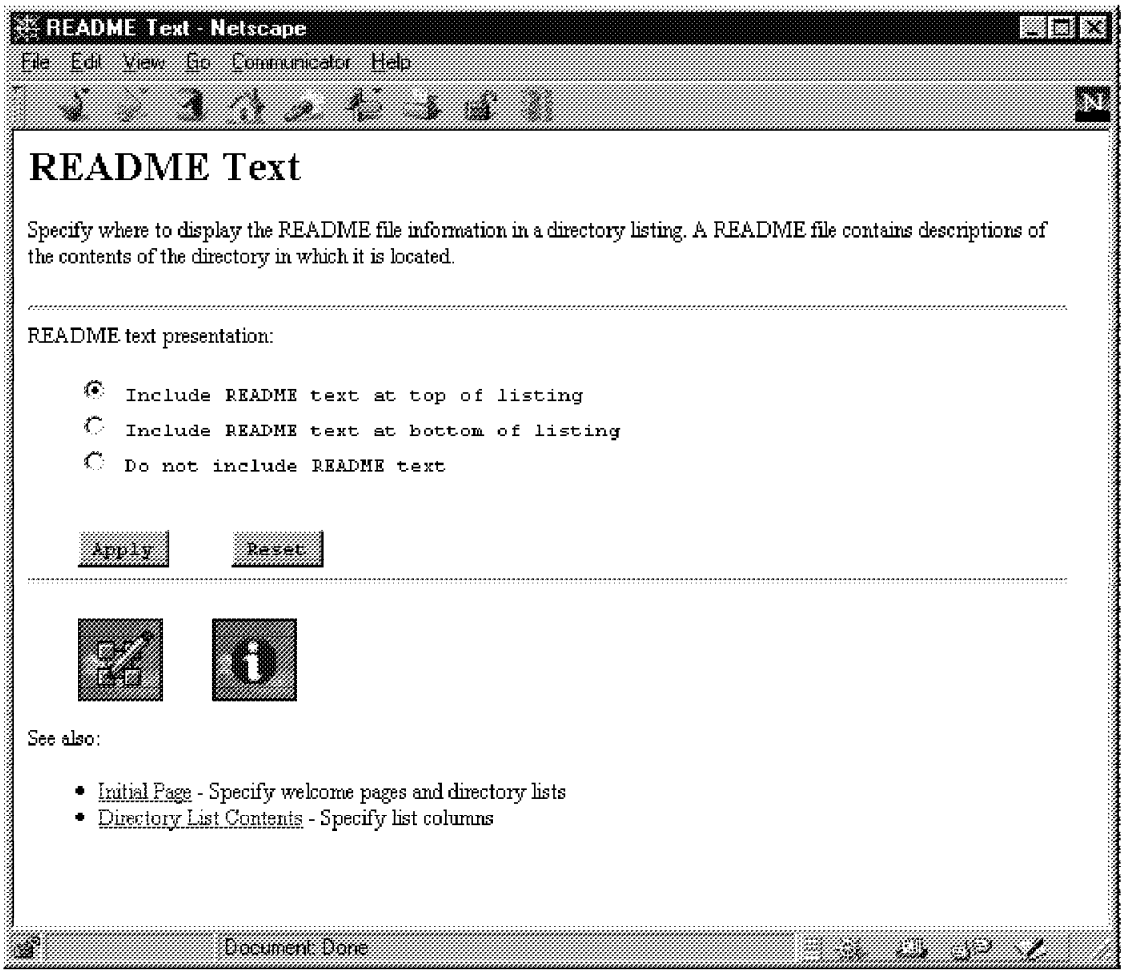


Figure 47. Readme Text

To define use of README information on directory listings:

1. Select one:
   - **Include README text at top of listing** - display README information at the top of the directory listings.
   - **Include README text at bottom of listing** - display README information at the bottom of directory listings.
   - **Do not include README text** - do not display README information on directory listings.
2. Click **Apply** to update the server with the changes you made to the form, or click **Reset** to return to the values that were on the form before you made the changes.

```
                HTTP Configuration File

          DirReadme       Top
```

Figure 48. README Text Added to HTTP Configuration File

Figure 48 shows the results of applying the settings in the form shown in Figure 47 on page 71.

6.5 Logging

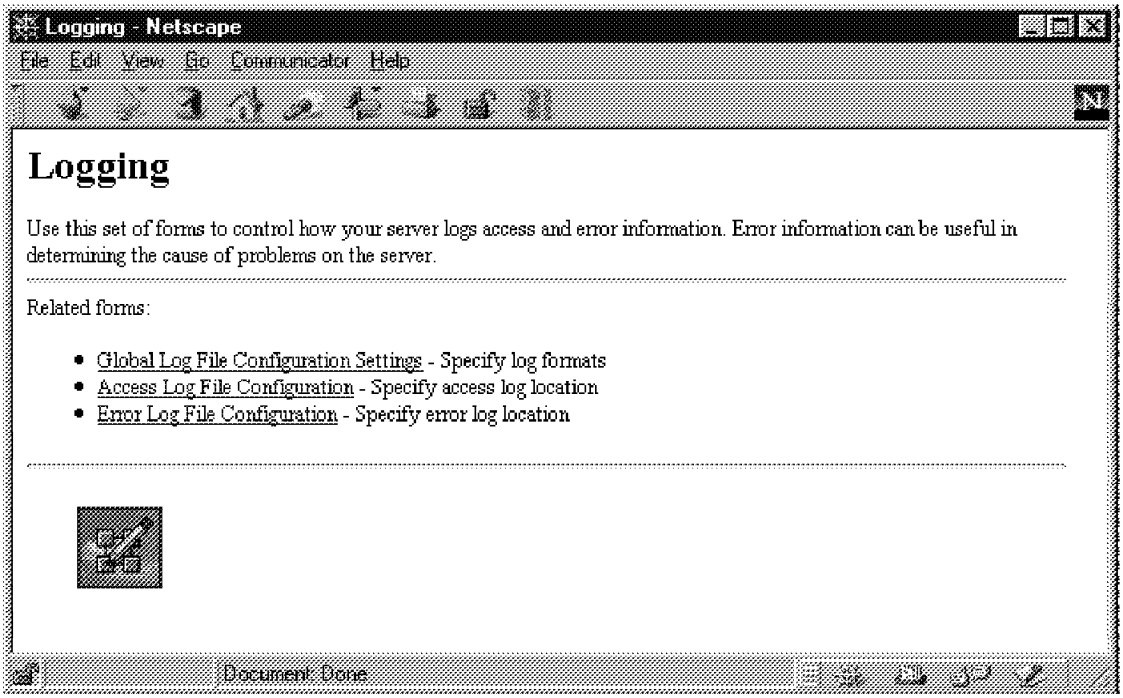


Figure 49. Logging

Through this form, we define the following configurations:

- Global log file configuration settings
- Access log file configuration
- Error log file configuration

6.5.1 Global Log File Configuration Settings

Use this form to:

- Choose a time stamp for the logs.
- Specify a log file format.

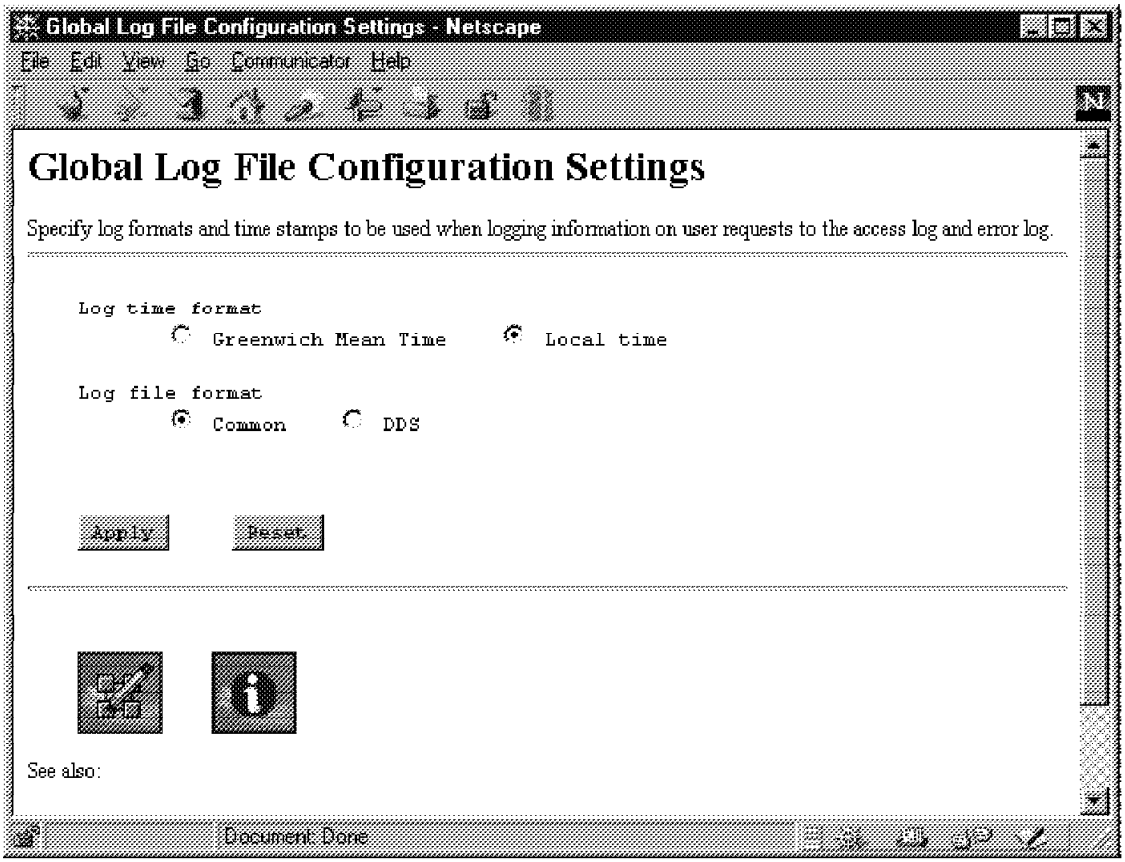


Figure 50. Global Log File Configuration Settings

To specify global log settings:

1. Select one of the following:
   - **Greenwich Mean Time** - use Greenwich Mean Time on time stamps. (System value QUTCOFFSET is used to calculate the GMT from local time.)
   - **Local time** - use local time on time stamps.
2. Select one of the following file formats:
   - **Common** - write logs in the format used by most Web servers. In this format, the log file can be FTPd or otherwise transferred to a PC for analysis.
   - **DDS** - access and error log files are to be stored in QUSRSYS. In this format, the log file can be analyzed with an AS/400 query or an application. For further information, see the *Webmaster's Guide*, GC41-5434.

   The log format used can be overriden using instance start-up values. See 6.11, “Start TCP/IP Server and End TCP/IP Server Commands” on page 102.

3. Click **Apply** to update the server with the changes you made to the form, or click **Reset** to return to the values that were on the form before you made the changes.

```
                HTTP Configuration File

          LogTime       localtime
          LogFormat       Common
```

Figure 51. Global Log File Settings Added to HTTP Configuration File

Figure 51 shows the results of applying the settings in the form shown in Figure 50 on page 73.

6.5.2 Access Log File Configuration

Use this form to configure the server to log information based on host name or IP address and to specify where to log access information.

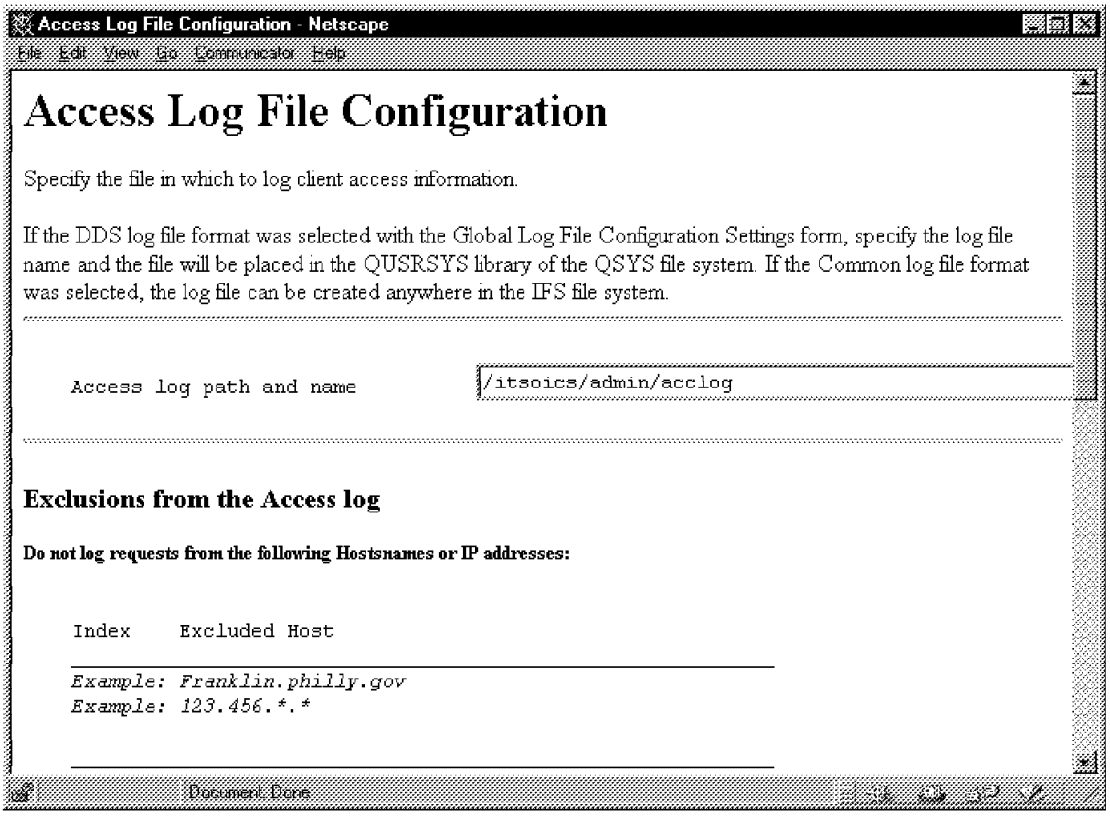


Figure 52. Access Log File Configuration - Basic

To specify basic access log settings:

1. Fill in the **Access log path and name** where you want to place access logging information.
   - If you selected **Common** for **Log file format** on the Global Log File Configuration Settings form (see Figure 50 on page 73), you can choose whether you want to store the access log file in the QUSRSYS library as an EBCDIC file or the IFS file system as an ASCII file.

Note: The access logs and error logs must be placed in the same file system.

- For the QSYS file system, the file is stored in the QUSRSYS library. Enter only the file name to use for the access log. The file is stored in the QUSRSYS library. For example:

  ```
  acclog
  ```

- For the IFS file system, enter the path of a valid directory and the file name. The directory must be created before starting the server. For example, enter:

  ```
  /http/logs/accesslog
  ```

- If you selected **DDS** for **Log file format** on the Global Log File Configuration Settings form, you must store the access log file in the QSYS file system. The file is stored in the QUSRSYS library. Enter only the file name to use for the access log, for example:

  ```
  acclog
  ```

The log file name can be overriden through instance parameters (see Figure 35 on page 56) or instance start-up values (see 6.11, “Start TCP/IP Server and End TCP/IP Server Commands” on page 102).

The system generates a file extension or member name in the format of q*cyymmdd* where:

- *c* is the century indicator (0 for pre-2000, 1 for post-2000).
- *yy* is the year indicator.
- *mm* is the month indicator.
- *dd* is the day indicator.

Scroll down for Access Log File filtering.

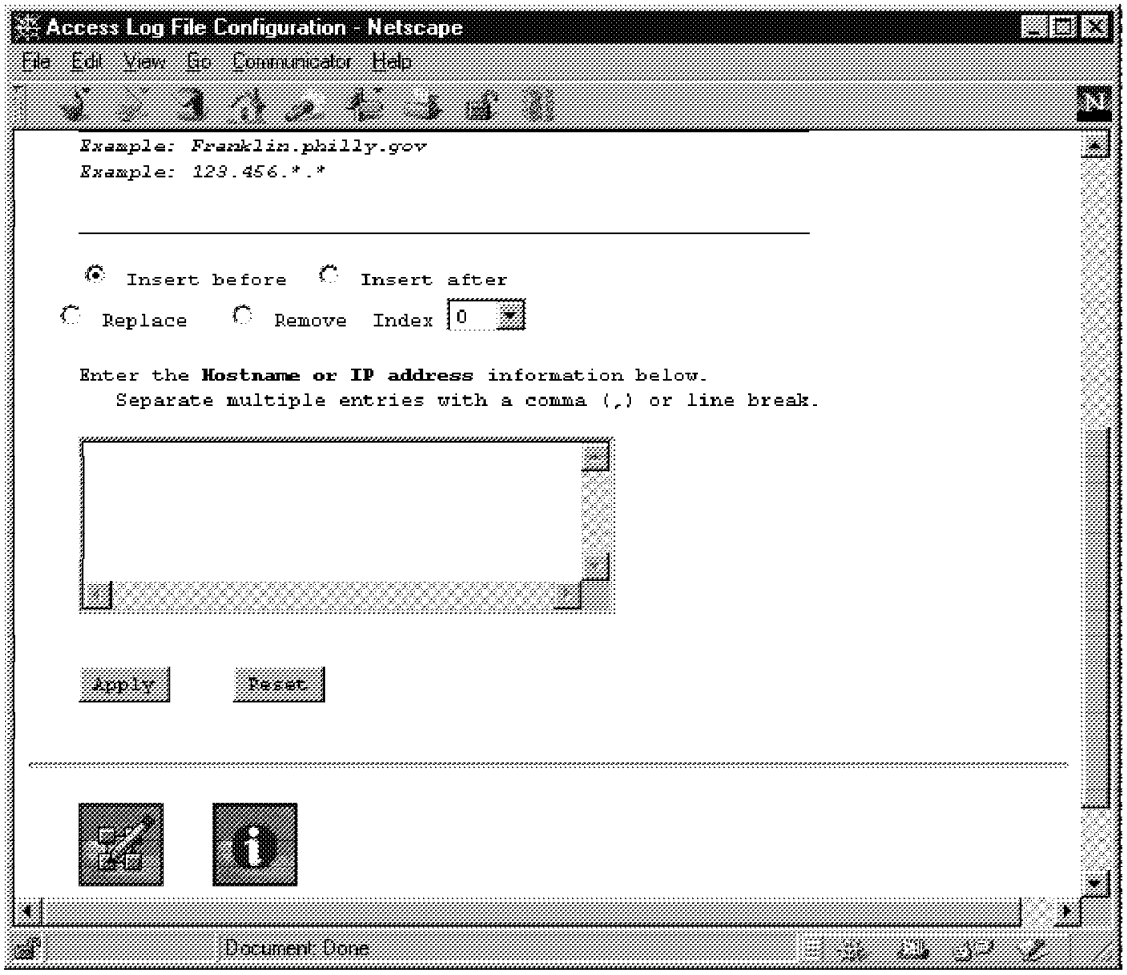


Figure 53. Access Log File Configuration - Filtering

You can limit the information logged by excluding host names or IP addresses.

To exclude host names or IP addresses from the access log:

1. Select either **Insert before** or **Insert after**.
2. Select an **Index number**.
   - Your choices in step 1 and step 2 indicate the position you want the item to have in the list. For example, if you select **Insert before** and **Index 2**, the item is second in the list. If you select **Insert after** and **Index 4**, the item is fifth in the list.
   - Fill in the field **Enter host name or IP address information**. Specify the host name or IP address you want to exclude. Separate multiple entries with a comma or line break.

To replace host names or IP addresses in the access log list:

1. Select **Replace**.
2. Select the **Index number** of the item you want to replace.
3. Fill in the following fields:

- **Enter the host name or IP address information -** specify the host name or IP address you want to replace. Separate multiple entries with either a comma or line break.

To remove host names or IP addresses from the access log list:

1. Select **Remove**.
2. Select the **Index number** of the item you want to remove.

Click **Apply** to update the server with the changes you made to the form, or click **Reset** to return to the values that were on the form before you made the changes.

```
                HTTP Configuration File

          AccessLog       /itsoics/admin/acclog
```

Figure 54. Access Log File Configuration Added to HTTP Configuration File

Figure 54 shows the results of applying the settings in the form shown in Figure 52 on page 74 and Figure 53 on page 76.

The following figure shows an example of a DDS format access log file:

```
                                Display Report
                                                Report width . . . . . :     546
Position to line  . . . . .  _____      Shift to column  . . . . . .  _____
Line   ....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
       Access                           Clients                          URL
       time                             location                         Address
000001 [29/Oct/1997:08:45:43 +0600]  9.5.100.110                      "GET /downlo
000002 [29/Oct/1997:08:46:02 +0600]  9.5.100.110                      "GET /downlo
000003 [29/Oct/1997:08:46:43 +0600]  9.5.100.110                      "GET /downlo
000004 [29/Oct/1997:08:58:22 +0600]  9.5.100.110                      "GET /downlo
000005 [29/Oct/1997:09:01:49 +0600]  9.5.100.112                      "GET /downlo
000006 [29/Oct/1997:09:03:15 +0600]  9.5.100.112                      "GET / HTTP/
000007 [29/Oct/1997:09:03:17 +0600]  9.5.100.112                      "GET /gifs/w
000008 [29/Oct/1997:09:03:17 +0600]  9.5.100.112                      "GET /gifs/a
000009 [29/Oct/1997:09:03:17 +0600]  9.5.100.112                      "GET /applet
000010 [29/Oct/1997:09:03:17 +0600]  9.5.100.112                      "GET /gifs/s
000011 [29/Oct/1997:09:03:17 +0600]  9.5.100.112                      "GET /applet
000012 [29/Oct/1997:11:51:51 +0600]  9.5.100.109                      "GET / HTTP/
000013 [29/Oct/1997:11:51:52 +0600]  9.5.100.109                      "GET /gifs/a
000014 [29/Oct/1997:11:51:52 +0600]  9.5.100.109                      "GET /gifs/s
000015 [29/Oct/1997:11:51:52 +0600]  9.5.100.109                      "GET /gifs/w
                                                                        More...
F3=Exit      F12=Cancel      F19=Left      F20=Right      F21=Split
```

6.5.3 Error Log File Configuration

Specify where to log error information.

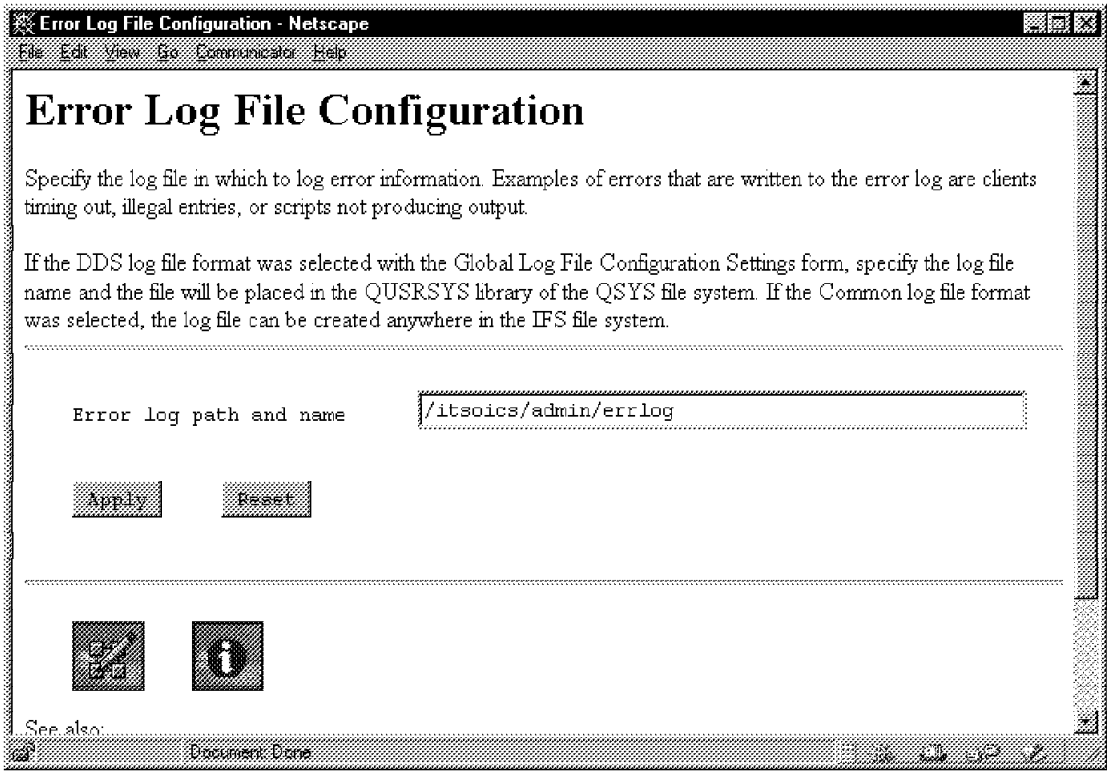


Figure 55. Error Log File Configuration

To specify basic error log settings

1. Fill in the following values:
   - **Error log path and name** - specify where to place error logging information.
     - If you selected **Common** for **Log file format** on the Global Log File Configuration Settings form (see Figure 50 on page 73), you can choose whether you want to store the error log file in the QSYS or the IFS file system.

       Note: The access logs and error logs must be placed in the same file system.

       - For the QSYS file system, the file is stored in the QUSRSYS library. Enter only the file name to use for the error log. The file is stored in the QUSRSYS library. For example:

         ```
         errlog
         ```

       - For the IFS file system, enter the path of a valid directory and the file name. The directory must be created before starting the server. For example, enter the following command:

         ```
         /http/logs/errorlog
         ```

     - If you selected **DDS** for **Log file format** on the Global Log File Configuration Settings form, you must store the error log file in the

QSYS file system. The file is stored in the QUSRSYS library. Enter only the file name to use for the error log, for example:

```
errlog
```

The log file name can be overriden through instance parameters (see Figure 35 on page 56) or instance start-up values (see 6.11, “Start TCP/IP Server and End TCP/IP Server Commands” on page 102).

The system generates a file extension in the format of q*cyymmdd* where:

- *c* is the century indicator (0 for pre-2000, 1 for post-2000).
- *yy* is the year indicator.
- *mm* is the month indicator.
- *dd* is the day indicator.

2. Click **Apply** to update the server with the changes you made to the form, or click **Reset** to return to the values that were on the form before you made the changes.

```
               HTTP Configuration File

         ErrorLog       /itsoics/admin/errlog
```

Figure 56. Error Log File Configuration Added to HTTP Configuration File

Figure 56 shows the results of applying the settings in the form shown in Figure 55 on page 78.

The following figure shows an example of a DDS format error log file:

```
                              Display Report
                                                Report width . . . . . :     338
Position to line  . . . . .  ______      Shift to column  . . . . . .  _____
Line    ....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
        Error                           Error                Location
        time                            code
000001 [19/Aug/1997:08:52:23 +0000]  Forbidden by rule.    9.5.148.141
000002 [19/Aug/1997:09:42:01 +0000]  User not authenticat  9.5.148.141
000003 [19/Aug/1997:09:43:04 +0000]  User not authenticat  9.5.148.141
000004 [19/Aug/1997:09:54:36 +0000]  User not authenticat  9.5.148.140
000005 [19/Aug/1997:09:54:50 +0000]  User not authenticat  9.5.148.140
000006 [19/Aug/1997:09:55:07 +0000]  User not authenticat  9.5.148.140
000007 [19/Aug/1997:09:55:16 +0000]  User not authenticat  9.5.148.140
000008 [19/Aug/1997:09:55:27 +0000]  User not authenticat  9.5.148.140
000009 [19/Aug/1997:10:04:05 +0000]  User not authenticat  9.5.148.141
000010 [19/Aug/1997:10:22:43 +0000]  User not authenticat  9.5.148.143
000011 [19/Aug/1997:11:14:14 +0000]  User not authenticat  9.5.148.143
000012 [19/Aug/1997:11:14:58 +0000]  User not authenticat  9.5.148.143
000013 [19/Aug/1997:11:19:27 +0000]  User not authenticat  9.5.148.143
000014 [19/Aug/1997:11:23:00 +0000]  User not authenticat  9.5.148.143
000015 [19/Aug/1997:11:24:48 +0000]  User not authenticat  9.5.148.143
                                                                   More...
F3=Exit      F12=Cancel      F19=Left      F20=Right      F21=Split
```

6.6 Resource Mapping

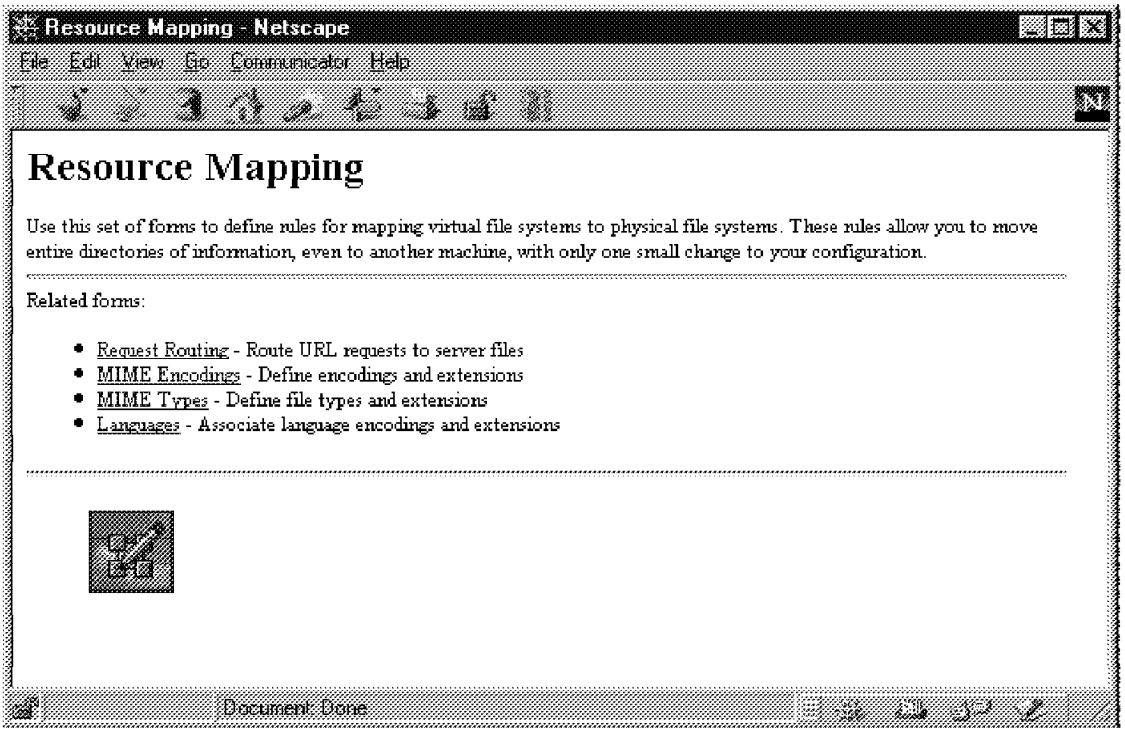


Figure 57. Resource Mapping

Through this form, we can define:

- Request routing
- MIME encoding
- MIME types
- Languages

6.6.1 Request Routing

Use this form to create or modify a set of rules for the server to use when mapping URL requests to a file.

You can use a list of mapping rules to define a virtual layout of the server's Web resources. The server uses the request routing to map requests to actual files. This allows the physical location of files and directories to change without any impact to requesters. Having a virtual layout also lets the server present resources from other file systems and servers as part of its own set of resources.

The server takes the requested URL and processes it through the list of mapping rules. The order of the rules within the list is important because processing starts from the top of the list and continues down. Processing ends when the request is accepted, rejected, or redirected to another server. It is possible for a map action to change the URL during the processing. The server then uses the new URL for processing subsequent rules.

You can specify a different set of mapping rules for the server to use depending on the address a request comes in on.

For example, you might want to specify that a request beginning with /cgi-bin/ received on address 9.67.106.79 is mapped to the /QSYS.LIB/CUSTA.LIB/* directory; and the same request received on an address matching the template 9.83.* is mapped to the /QSYS.LIB/CUSTB.LIB/* directory. Remember, CGI programs can reside only in the QSYS file system.

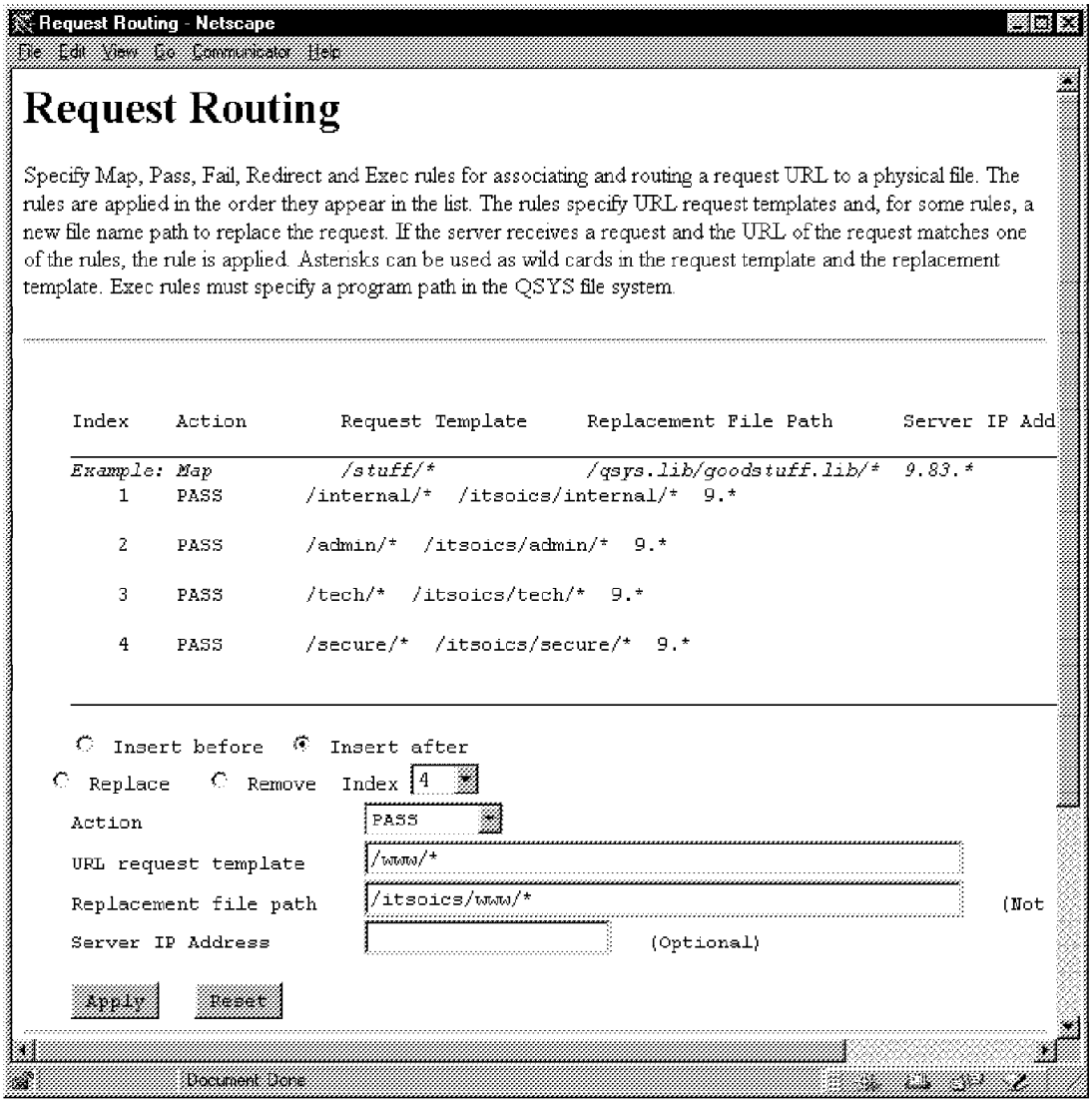


Figure 58. Request Routing

Mapping rules contain the following parts:

- **Action** - the server recognizes the following mapping rule actions.
  - Map - the server changes the URL according to the Replacement File Path. The server then compares the new URL to the Request Template on subsequent rules.
  - Pass - the server accepts the request. The Replacement File Path is optional with the Pass action. If a Replacement File Path is used, the server uses it to change the URL after accepting the request. The server does not compare the URL to any subsequent rules.
  - Fail - the server rejects the request. The server does not compare the URL to any subsequent rules.

Important

Be careful when using the fail action. The request template field is case sensitive. With the exception of QOpenSys file system, the AS/400 file is not case sensitive. It is easier to secure objects using the Pass directive than the Fail. This is because you need to add a Fail statement for every possible combination of upper and lower case letters in the object you are trying to secure.

Example

You want to refuse access to object *secure.htm*. Considering case sensitivity, you add two Fail statements as shown in the following example:

```
Fail  /admin/secure.htm
Fail  /admin/SECURE.HTM
```

You assume that you are secure since both upper and lower case possibilities have been covered. It is, however, possible to access the document using a URL such as:

```
http://ServerName/admin/SeCuRe.htm
```

Since the value SeCuRe.htm does not match either of the Fail statement templates, the request does not fail and the document is accessed.

We recommend using specific *Pass* statements and maybe, as an additional measure, a generic Fail /* ″catch-all″ at the bottom of the configuration file, although it should not be necessary if the Pass statements are explicit enough as requests should fail by default.

  - Exec - the server accepts the request and runs a CGI script program identified by the Replacement File Path. The server does not compare the URL to any subsequent rules. Selecting Exec enables CGI programs to run. Unless CGI programs are enabled by selecting Exec, the server will not honor a request to run a CGI program. Selecting Exec can enable all the CGI programs in a library or it can enable only specific programs within various libraries.
  - Redirect - the server sends the request to another server identified by the Replacement File Path. The Replacement File Path must contain a full URL. The server does not notify the requester that the request is actually being answered by another server. The server does not compare the URL to any subsequent rules.

- **Request Template** - the server compares requested URLs to the Request Templates in the mapping rules. A Request Template is required for each Action. The Request Template can contain the asterisk (*) wildcard character.

  Important!

  A request template of /* when associated with a Pass or Exec gives access to all server resources and is, therefore, not recommended.

- **Replacement File Path** - when a URL matches the Request Template, the server changes the URL according to the Replacement File Path. Replacement File Path is required for the Map, Exec, and Redirect directives; it is optional for the Pass directive; it is not used for the Fail action. If the

Request Template uses a wildcard, the Replacement File Path can also use a wildcard. If a wildcard is present on both, the part of the URL matching the wildcard is used as is. The Exec directive requires a wildcard at the end of both the Request Template and Replacement File Path. The part of the Replacement File Path before the wildcard identifies the path where the CGI script program is located. The part of the URL that matches the wildcard is the name of the CGI script program.

Table 4. Examples of Replacement File Paths

Request Template	Replacement File Path	Incoming URI	URI after Replacement	
/internal/*	/itsoics/internal/*	/internal/inthome.htm	/itsoics/internal/inthome.htm	
/cgi/*	/QSYS.LIB/WEB.LIB/*.pgm	/cgi/order	/QSYS.LIB/WEB.LIB/order.pgm	1
/secret/*	/itsoics/www/notfound.htm	/secret/moresecrets.htm	/itsoics/www/notfound.htm	
/*	/itsoics/www/home.htm	/whatever/page1.htm	/itsoics/www/home.htm	2

1 For the program to be executed, the directive has to be Exec or another Exec directive is needed for Request Template /QSYS.LIB/WEB.LIB/*.

2 If used, this is the last Pass directive to force all remaining requests to a valid page. **Do not** use Request Template /* without a Replacement File Path.

To insert a new mapping rule:

1. Select either **Insert before** or **Insert after**.
2. Select an **Index number**.
   - Your choices in step 1 and step 2 indicate the position you want the item to have in the list. For example, if you select **Insert before** and **Index 2**, the item is second in the list. If you select **Insert after** and **Index 4**, the item is fifth in the list.
3. Enter the new mapping rule by selecting an **Action** and filling in the **URL request template** and **Replacement file path** fields. See the Mapping Rules earlier in this section for a description.
4. Optionally specify the server IP address to associate with the mapping rule.
5. Click **Apply** to update the server with the changes you made to the form, or click **Reset** to return to the values that were on the form before you made the changes.

To replace a mapping rule:

1. Select **Replace**.
2. Select the **Index number**.
3. Enter the new mapping rule by selecting an **Action** and filling in the **URL request template** and **Replacement file path** fields. See the Mapping Rules earlier in this section for a description of the parts of a mapping rule.
4. Optionally specify the server IP address to associate with the mapping rule.
5. Click **Apply** to update the server with the changes you made to the form, or click **Reset** to return to the values that were on the form before you made the changes.

To remove a mapping rule:

1. Select **Remove**.

2. Select the **Index number**.
3. Click **Apply** to update the server with the changes you made to the form, or click **Reset** to return to the values that were on the form before you made the changes.

```
               HTTP Configuration File

         PASS       /internal/*  /itsoics/internal/*  9.*
         PASS       /admin/*  /itsoics/admin/*  9.*
         PASS       /tech/*  /itsoics/tech/*  9.*
         PASS       /secure/*  /itsoics/secure/*  9.*
         PASS       /www/*  /itsoics/www/*
```

Figure 59. Request Routing Added to HTTP Configuration File

Figure 59 shows the results of applying the settings in the form shown in Figure 58 on page 81.

6.6.2 MIME Encodings

Use this form to create or modify a list of file extensions that you want to bind to Multipurpose Internet Mail Extension (MIME) encodings. The contents of files are encoded using methods defined by the MIME extension to the Internet mail standard. You can use a list to associate file extensions with MIME encodings. The server considers any file with a file extension matching an entry in the list to be encoded with the associated MIME encoding method.

You can also define whether you want the server to differentiate between upper case and lower case letters on file extensions.

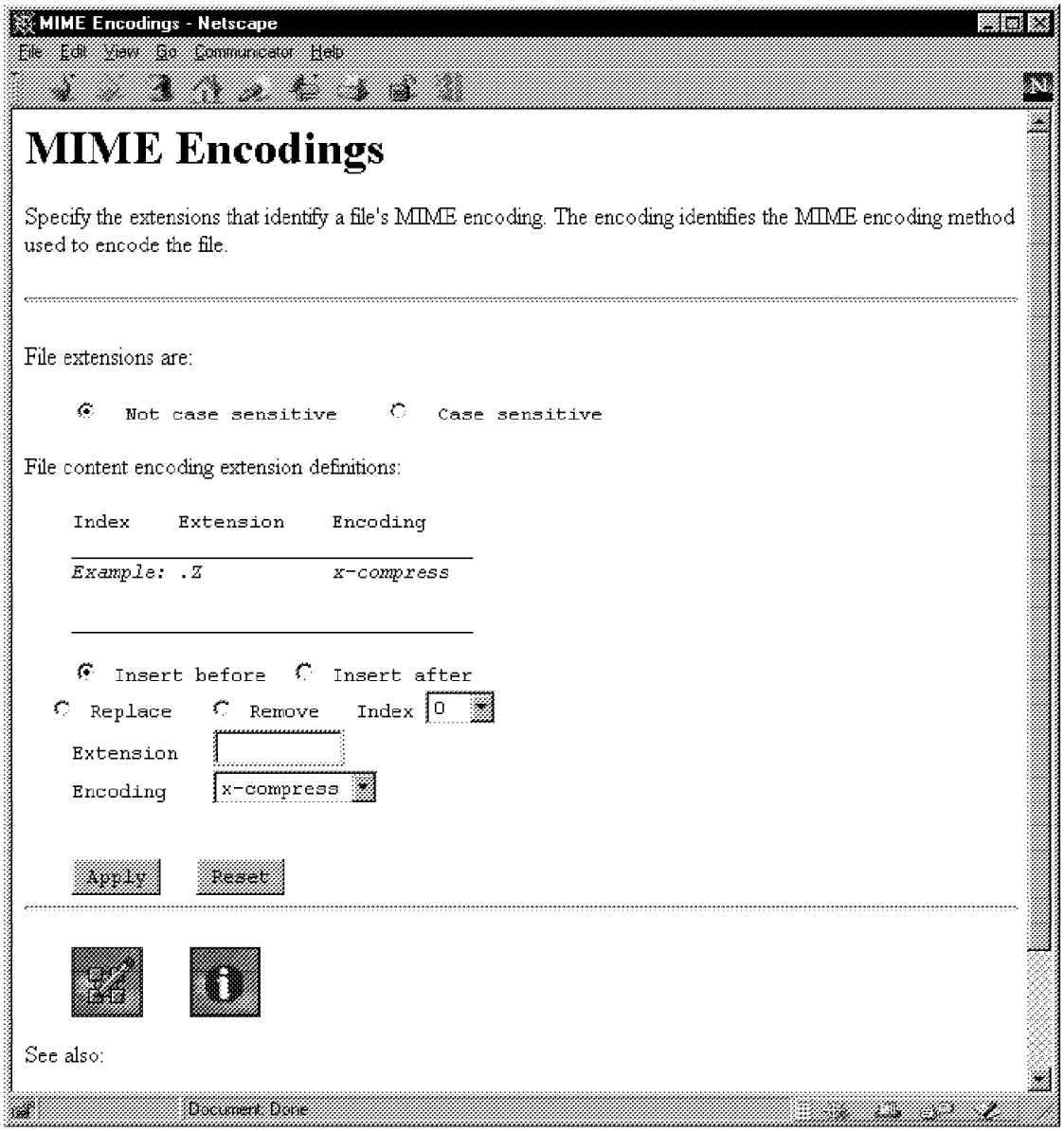


Figure 60. MIME Encodings

To set case sensitivity for file extensions:

1. Select one:

- **Not case sensitive** - the server does not differentiate between upper case and lower case letters on file extensions.
- **Case sensitive** - the server differentiates between upper case and lower case letters on file extensions.

  Note: The setting for case sensitivity applies to all file extensions. If you change the setting on this form, the change also applies to the MIME Types form and the Languages form.

To insert a new entry:

1. Select either **Insert before** or **Insert after**.
2. Select an **Index number**.
   - Your choices in step 1 and step 2 indicate the position you want the item to have in the list. For example, if you select **Insert before** and **Index 2**, the item is second in the list. If you select **Insert after** and **Index 4**, the item is fifth in the list.
3. Fill in the following fields:
   - **Extension** - the file extension you want to associate with a MIME encoding. Include the leading period (.) with the extension.
   - **Encoding** - the MIME encoding method you want to bind to the file extension.

To replace an entry:

1. Select **Replace**.
2. Select the **Index number** of the item you want to replace.
3. Fill in the following fields:
   - **Extension** - the file extension you want to associate with a MIME encoding. Include the leading period with the extension.
   - **Encoding** - the MIME encoding method you want to bind to the file extension.

To remove an entry:

1. Select **Remove**.
2. Select the **Index number** of the item you want to remove.

Click **Apply** to update the server with the changes you made to the form, or click **Reset** to return to the values that were on the form before you made the changes.

6.6.3 MIME Types

Use this form to create or modify a list of file extensions that you want to bind to Multipurpose Internet Mail Extension (MIME) content types/subtypes. File contents are packaged using methods defined by the MIME extension to the Internet mail standard. You can use a list to associate file extensions with MIME types/subtypes. The server considers any file with an extension matching an entry in the list to be packaged with the associated MIME type/subtype.

Note: Default file extensions for MIME types are described in the *Webmaster's Guide*, GC41-5434. The examples on the form are **not** default values.

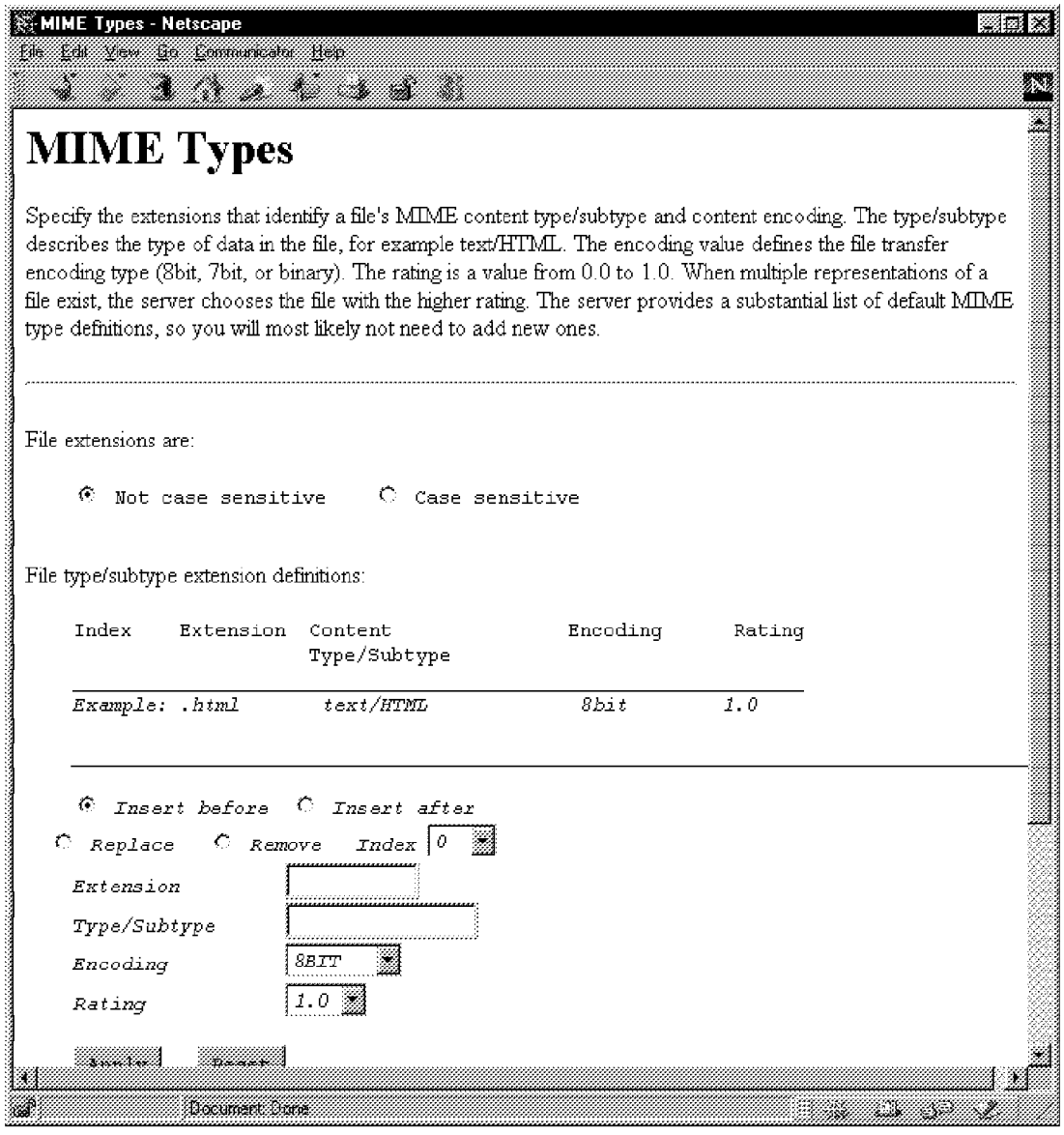


Figure 61. MIME Types

There are two special extension patterns you can include in the list:

. Matches all file names that contain a period and have not been matched by other entries in the list.

***** Matches all file names that do not contain a period and have not been matched by other entries in the list.

You can also define whether you want the server to differentiate between upper case and lower case letters on file extensions.

To set case sensitivity for file extensions:

1. Select one of the following cases:
   - **Not case sensitive** - the server does not differentiate between upper case and lower case letters on file extensions.
   - **Case sensitive** - the server differentiates between upper case and lower case letters on file extensions.

   Note: The setting for case sensitivity applies to all file extensions. If you change the setting on this form, the change also applies to the MIME Encoding form and the Languages form.
2. Click **Apply** to update the server with the changes you made to the form, or click **Reset** to return to the values that were on the form before you made the changes.

To insert a new entry:

1. Select either **Insert before** or **Insert after**.
2. Select an **Index number**.
   - Your choices in step 1 and step 2 indicate the position you want the item to have in the list. For example, if you select **Insert before** and **Index 2**, the item is second in the list. If you select **Insert after** and **Index 4**, the item is fifth in the list.
3. Fill in the following fields:
   - **Extension** - the file extension you want to associate with a MIME type/subtype. Include the leading period with the extension.
   - **Content Type/Subtype** - the MIME type/subtype you want to bind to the extension.
   - **Encoding** - the MIME type encoding is usually 8 bit, 7 bit, or binary.
   - **Rating** - a number between 0.1 and 1.0 that indicates the relative value of the MIME type. The server uses the rating when there are multiple representations of a file to choose from. The server selects the file with the extension associated with the highest Rating value. Using this field is optional.
4. Click **Apply** to update the server with the changes you made to the form, or click **Reset** to return to the values that were on the form before you made the changes.

To replace an entry:

1. Select **Replace**.
2. Select the **Index number** of the item you want to replace.
3. Fill in the following fields:
   - **Extension** - the file extension you want to associate with a MIME type/subtype. Include the leading period with the extension.
   - **Content Type/Subtype** - the MIME type/subtype you want to bind to the extension.

- **Encoding** - the MIME type encoding is usually 8 bit, 7 bit, or binary.
- **Rating** - a number between 0.1 and 1.0 that indicates the relative value of the MIME type. The server uses the rating when there are multiple representations of a file to choose from. The server selects the file with the extension associated with the highest Rating value.

4. Click **Apply** to update the server with the changes you made to the form, or click **Reset** to return to the values that were on the form before you made the changes.

To remove an entry:

1. Select **Remove**.
2. Select the **Index number** of the item you want to remove.
3. Click **Apply** to update the server with the changes you made to the form, or click **Reset** to return to the values that were on the form before you made the changes.

6.6.4 Languages

Use this form to:

- Create or modify a list of file extensions that you want to bind to languages. If you intend for the server to have files in multiple languages, you can use a list to associate file extensions with the languages you want to use. The server considers any file with an extension matching an entry in the list to be in the associated language.
- Define whether you want the server to differentiate between upper case and lower case letters on file extensions.

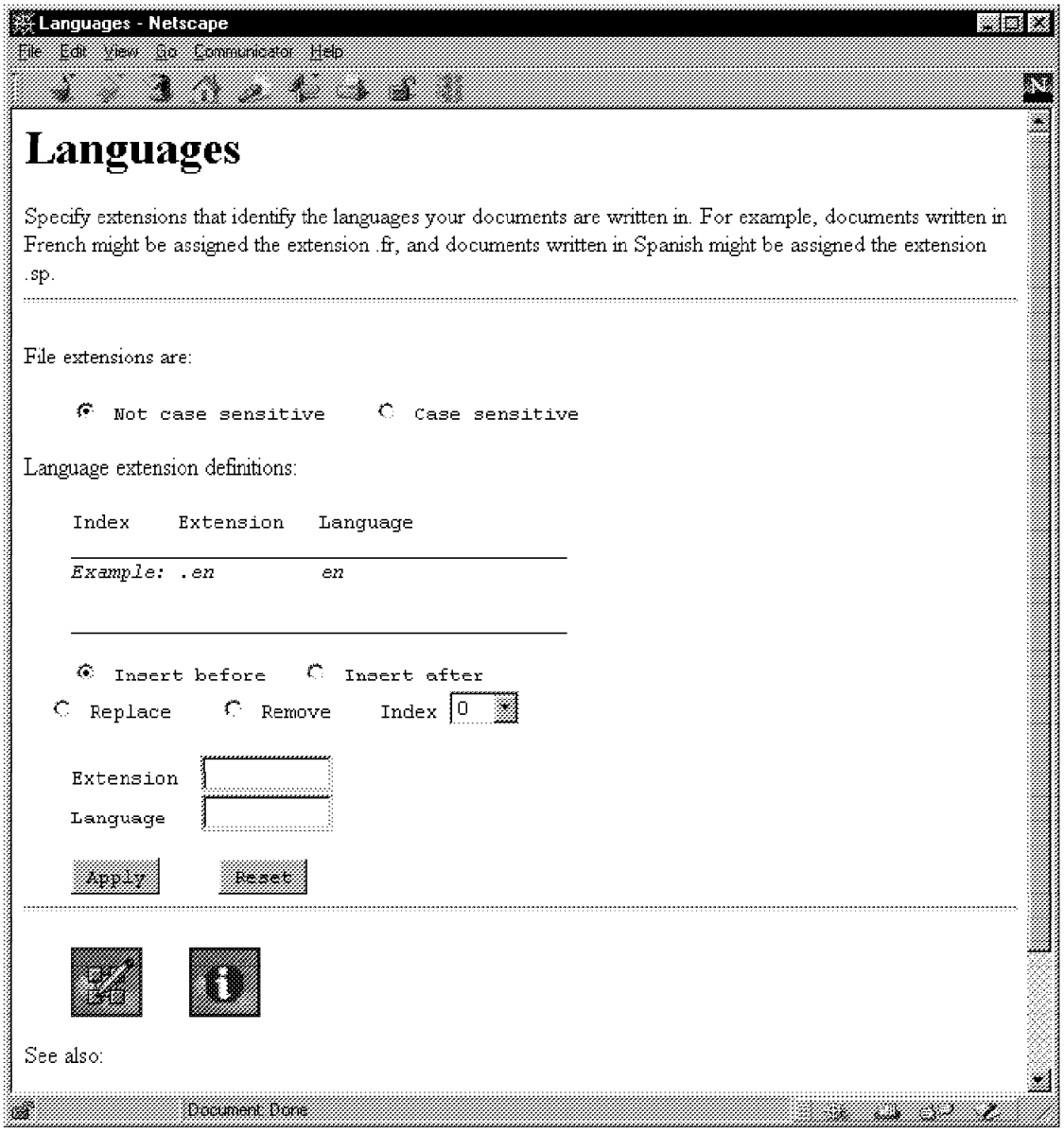


Figure 62. Languages

To set case sensitivity for file extensions:

1. Select one:

- **Not case sensitive** - the server does not differentiate between upper case and lower case letters on file extensions.
- **Case sensitive** - the server differentiates between upper case and lower case letters on file extensions.

  Note: The setting for case sensitivity applies to all extensions. If you change the setting on this form, the change also applies to the MIME Types form and the MIME Encodings form.

2. Click **Apply** to update the server with the changes you made to the form, or click **Reset** to return to the values that were on the form before you made the changes.

To insert a new entry:

1. Select either **Insert before** or **Insert after**.
2. Select an **Index number**.
   - Your choices in step 1 and step 2 indicate the position you want the item to have in the list. For example, if you select **Insert before** and **Index 2**, the item is second in the list. If you select **Insert after** and **Index 4**, the item is fifth in the list.
3. Fill in the following fields:
   - **Extension** - the file extension you want to associate with a language. Include the leading period with the extension.
   - **Language** - the name of the language you want to bind to the extension.
4. Click **Apply** to update the server with the changes you made to the form, or click **Reset** to return to the values that were on the form before you made the changes.

To replace an entry:

1. Select **Replace**.
2. Select the **Index number** of the item you want to replace.
3. Fill in the following fields:
   - **Extension** - the file extension you want to associate with a language. Include the leading period with the extension.
   - **Language** - the name of the language you want to bind to the extension.
4. Click **Apply** to update the server with the changes you made to the form, or click **Reset** to return to the values that were on the form before you made the changes.

To remove an entry:

1. Select **Remove**.
2. Select the **Index number** of the item you want to remove.
3. Click **Apply** to update the server with the changes you made to the form, or click **Reset** to return to the values that were on the form before you made the changes.

6.7 Timeouts

Use this form to define the maximum amount of time the server is to spend performing various functions. The time limits prevent processes from becoming hung for indefinite periods of time.

The server has a default value for each function's timeout setting. The defaults are appropriate for handling most requests, but you may choose to change them because of specific needs. Specify a time using any combination of hours, mins (minutes), and secs (seconds).

Examples of valid timeout values:

```
15 mins
1 hour 30 mins
2 mins 45 secs
```

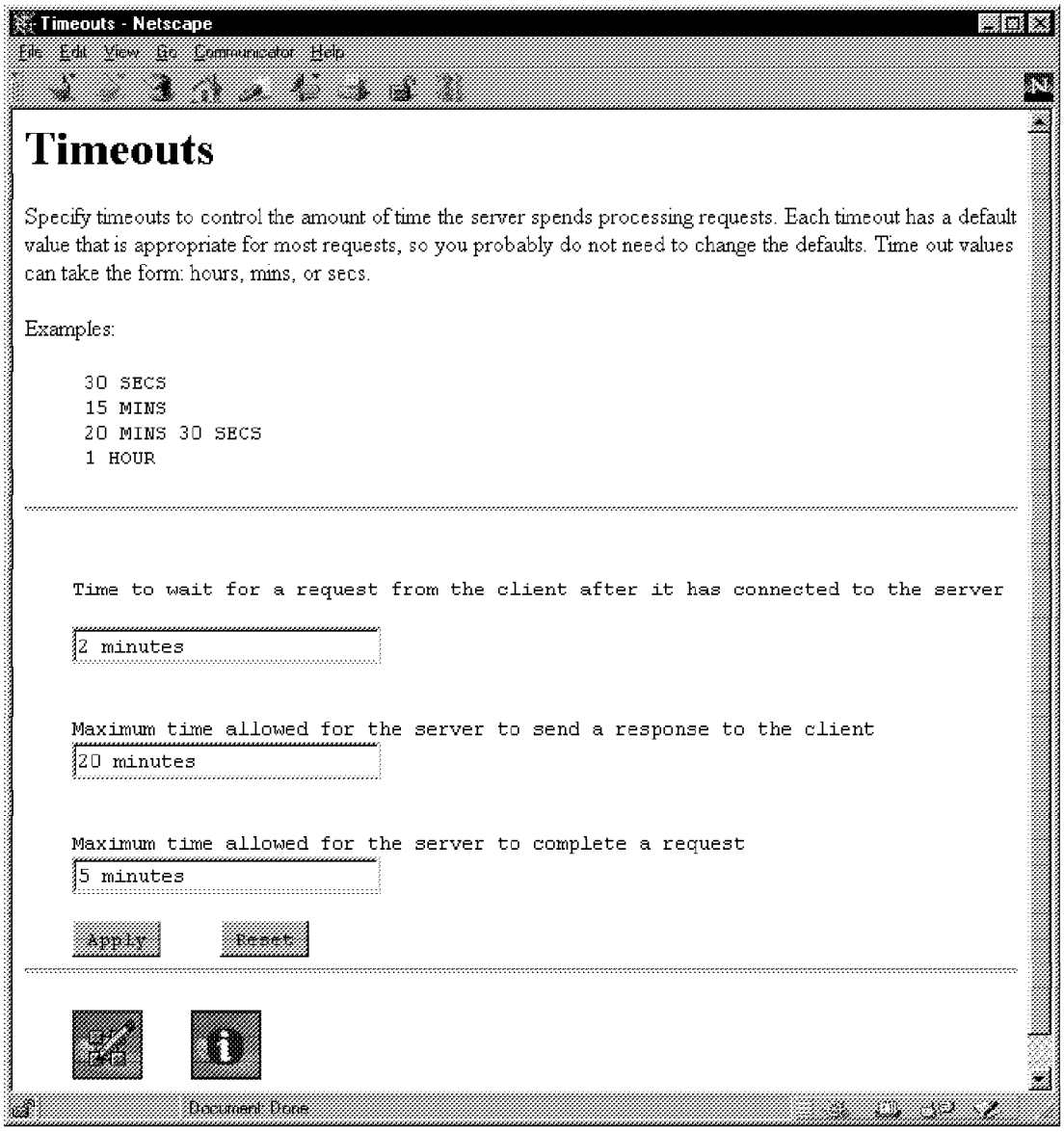


Figure 63. Timeouts

To change timeout values:

1. Fill in the following fields:
   - **Time to wait for a request from the client after it has connected to the server** - if a client connects to the server and then does not send a request within the amount of time specified in this field, the server drops the connection.
   - **Maximum time allowed for the server to send a response to the client** - if the server takes longer to send a response than the amount of time specified in this field, the server drops the connection to the client.
   - **Maximum time allowed for the server to complete a request** - if the server is requested to start a program and the program runs longer than the amount of time specified in this field, the server stops the program.
2. Click **Apply** to update the server with the changes you made to the form, or click **Reset** to return to the values that were on the form before you made the changes.

```
                HTTP Configuration File

          InputTimeOut       2 minutes
          OutputTimeOut       20 minutes
          ScriptTimeOut       5 minutes
```

Figure 64. Timeouts Settings Added to HTTP Configuration File

Figure 64 shows the results of applying the settings in the form shown in Figure 63 on page 92.

6.8 Methods

Use this form to select the Hypertext Transfer Protocol (HTTP) methods to be enabled for the server. Client requests to the server include a method field that indicates the action the server is to perform on the document or object the client is requesting. The client request is accepted only if the requested method is enabled on the server.

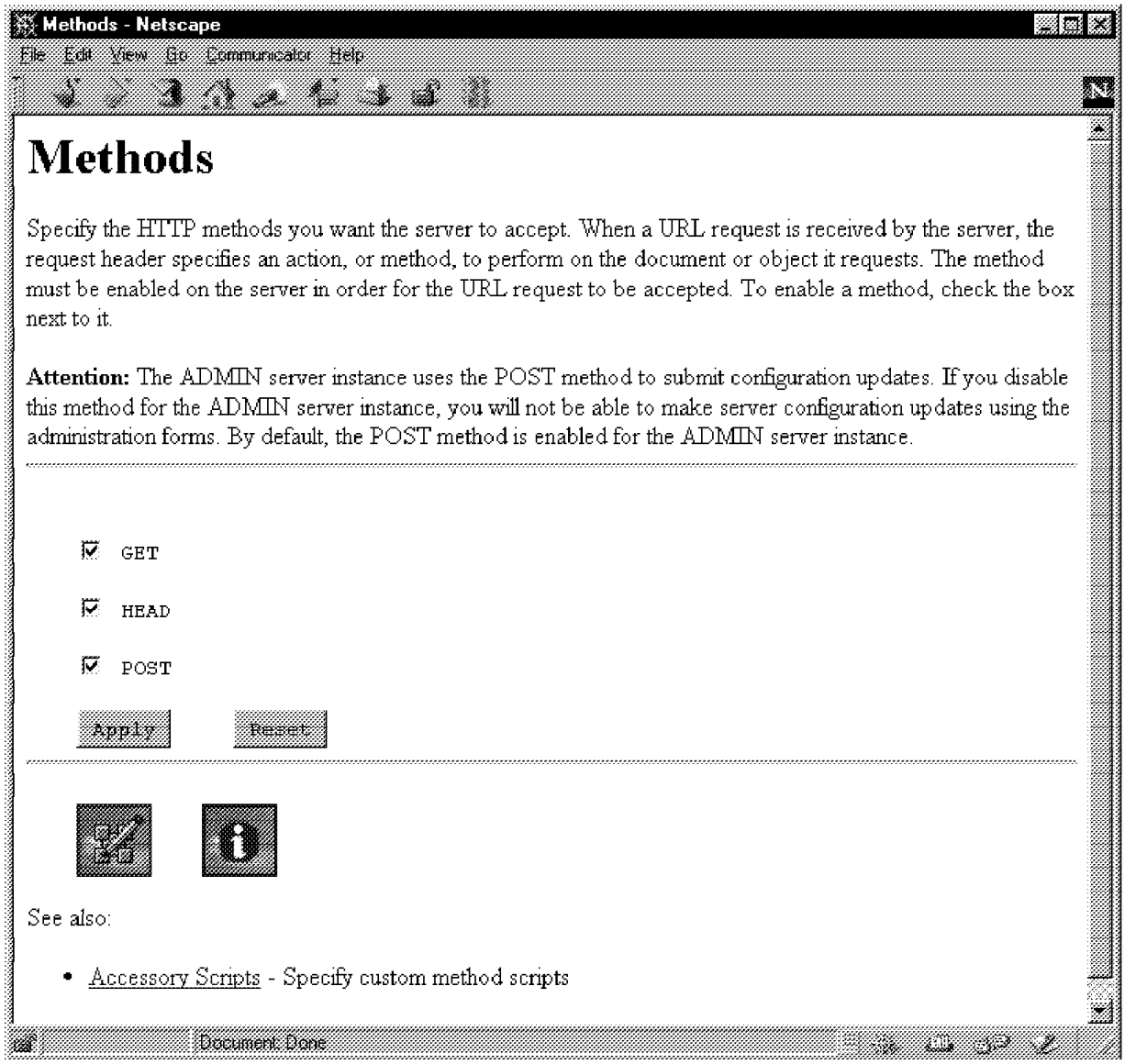


Figure 65. Methods

To enable or disable methods:

1. Check the boxes for the methods you want to enable. Uncheck the boxes for the methods you want to disable.

   The following descriptions show how the server responds to requests with each of the methods. The descriptions assume the method is enabled.

- **GET** - the server returns whatever data is identified by the URL. If the URL refers to an executable program, the server returns the output of the program.

  Use the following method to enable a CGI program to process the GET request:

  – Select Exec as the Action on the Request Routing form to enable CGI programs to run. Unless CGI programs are enabled by selecting Exec, the server will not honor a request to run a CGI program. Selecting Exec can enable all the CGI programs in a library, or it can enable only specific programs within various libraries.

  Use the following method to enable the serving of objects to the GET request:

  – Select Pass as the Action on the Request Routing form to enable object serving.

- **HEAD** - the server returns only an HTTP document header without the document body.

  Use the following method to enable the serving of objects to the HEAD request:

  – Select Pass as the Action on the Request Routing form to enable object serving.

- **POST** - POST is one form of CGI request. POST indicates that the inputs to the CGI program are passed to the CGI program in the standard input stream. To handle POST method requests submitted by a remote HTTP client, the server administrator needs to complete the following fields:

  a. Select **POST** on the Methods form to enable the POST method. (POST is disabled by default.)

  b. Use one of the following methods to enable a CGI program to process the POST request:

     – Select Exec as the Action on the Request Routing form to enable CGI programs to run. Unless CGI programs are enabled by selecting Exec, the server will not honor a request to run a CGI program. Selecting Exec can enable all the CGI programs in a library, or it can enable only specific programs within various libraries.

     – On the Accessory Scripts form, specify the path and file name of an accessory script to process POST requests not explicitly mapped to a program by the resource mapping rules. Specifying a script enables a default POST handling program that the server calls when both of the following conditions are true:

       - The CGI method is POST.

       - The CGI program named in the URL has not been enabled by a specifying the Exec on the Request Routing form.

2. Click **Apply** to update the server with the changes you made to the form, or click **Reset** to return to the values that were on the form before you made the changes.

```
                HTTP Configuration File

          Enable       GET
          Enable       POST
          Enable       HEAD
```

Figure 66. Timeouts Settings Added to HTTP Configuration File

Figure 66 shows the results of applying the settings in the form shown in Figure 65 on page 94.

6.9 Accessory Scripts

Use this form to define accessory script programs to process POST requests. Accessory script programs can be used to handle POST requests. The server first matches requests against its resource mapping rules. If the request does not explicitly map to a script program, the request is processed by the appropriate accessory script.

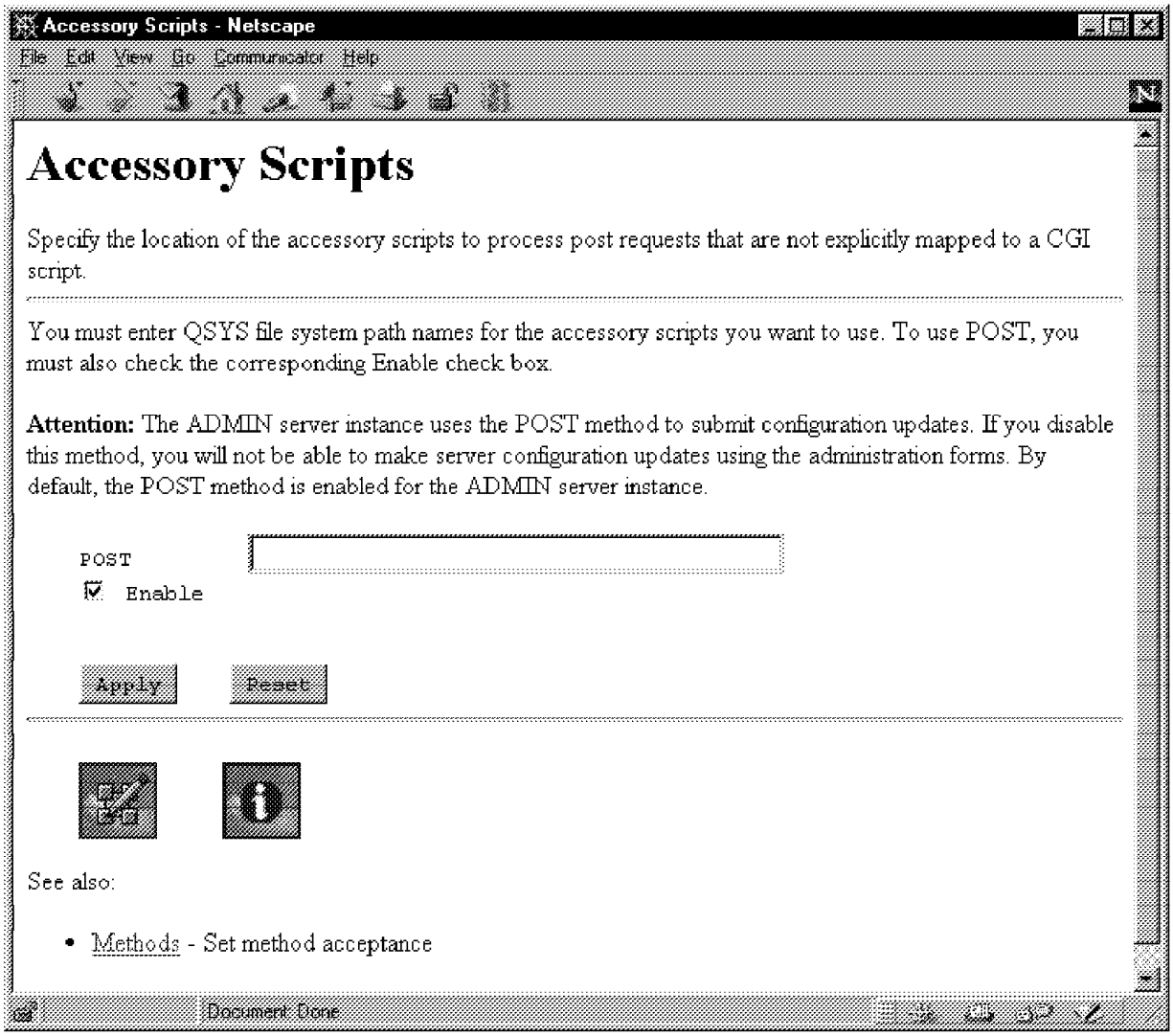


Figure 67. Accessory Scripts

To define accessory scripts:

1. Enter the absolute path and file names of the programs you want the server to use as accessory scripts.
2. Enter a name for each method you want to associate with an accessory script.
   - **POST** - the accessory script to process POST requests not explicitly mapped to a program by the resource mapping rules. Specifying a script enables a default POST handling program that the server calls when both of the following conditions are true:

- The CGI method is POST.
- The CGI program named in the URL has not been enabled by a specifying the Exec on the Request Routing form.

If an accessory script is not specified, such POST requests are rejected. Here is an example of a path and file name to enter in the **POST** field:

```
/QSYS.LIB/AS400CGI.LIB/WEBCG.PGM
```

3. Click **Apply** to update the server with the changes you made to the form, or click **Reset** to return to the values that were on the form before you made the changes.

6.10 Performance Settings

The only pure performance setting for the server that can be set through the Web browser configuration is the number of jobs for the server instance (see Section 6.10.1, “Jobs” on page 99).

The following tuning tips can improve server performance:

- Suppress DNS lookup (see Section 6.3, “Basic Settings” on page 61).
- Turn access logging off or suppress access logging from selected browsers (see Section 6.5, “Logging” on page 72).
- Turn error logging off (see Section 6.5, “Logging” on page 72).
- Order resource mapping templates (see Section 6.6, “Resource Mapping” on page 80).
- Minimize directory depth.
- Tell ICS if not using ACLs (see Section 8.4.12, “Access Control Lists” on page 151).
- Tell ICS if not using directory listings (see Section 6.4.3, “Directory List Contents” on page 69).
- Optimize directory listings, if using (see Section 6.4.3, “Directory List Contents” on page 69).
- Eliminate extraneous statements in configuration file.
- Secure operations are slower.

6.10.1 Jobs

Use this form to control the performance of the server by setting a range for active or available jobs.

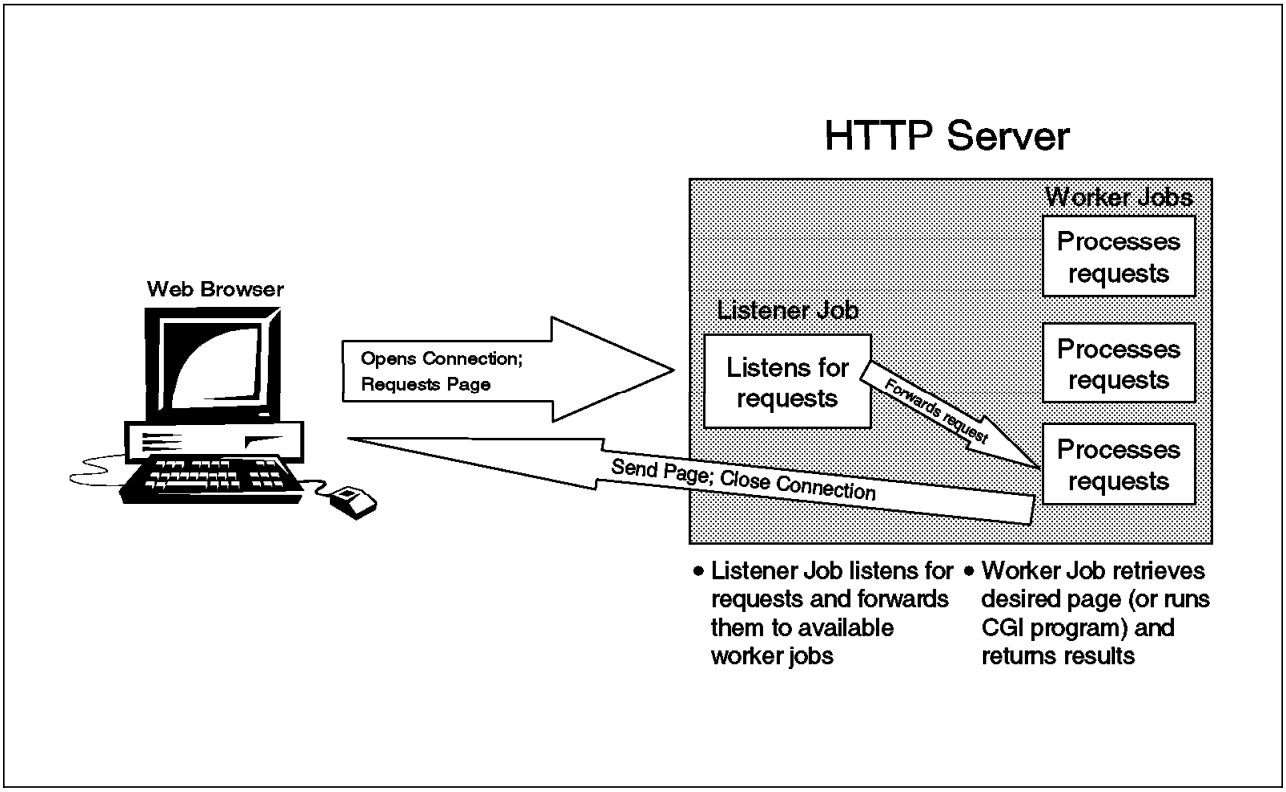


Figure 68. Listener and Worker Jobs

By having a pool of jobs available to the server, you eliminate the start-up time of a new worker job. This results in a faster response time for the client. Each time the server receives a request from a client, it attempts to pass that work off to an idle worker job. If there are no idle workers and the maximum number of worker jobs has not been reached, the server creates a new worker job and then passes the request to it.

```
                          Work with Active Jobs                    SYSTEM01
                                                                11/24/97  09:42:20
CPU %:    1.3     Elapsed time:   00:02:07     Active jobs:   229
Opt  Subsystem/Job  User        Type  CPU %  Function        Status
__     ITSO         QTMHHTTP    BCH      .0  PGM-QTMHHTTP     TIMW
__     ITSO         QTMHHTTP    BCI      .0                   DEQW
__     ITSO         QTMHHTTP    BCI      .0                   DEQW
__     ITSO         QTMHHTTP    BCI      .0                   SELW
__     ITSO         QTMHHTTP    BCI      .0                   SELW

                                                                          Bottom
===> ____________________________________________________________________________
F21=Display instructions/keys
```

Figure 69. Listener and Worker Jobs for Server Instance ITSO (WRKACTJOB JOB(ITSO))

The server first checks to see if any jobs are available. If so, the server uses available jobs to process the request. If not, the server has to start new jobs.

When the request finishes, the jobs it was using become idle. As long as idle jobs do not expire, they are available for the server to use again.

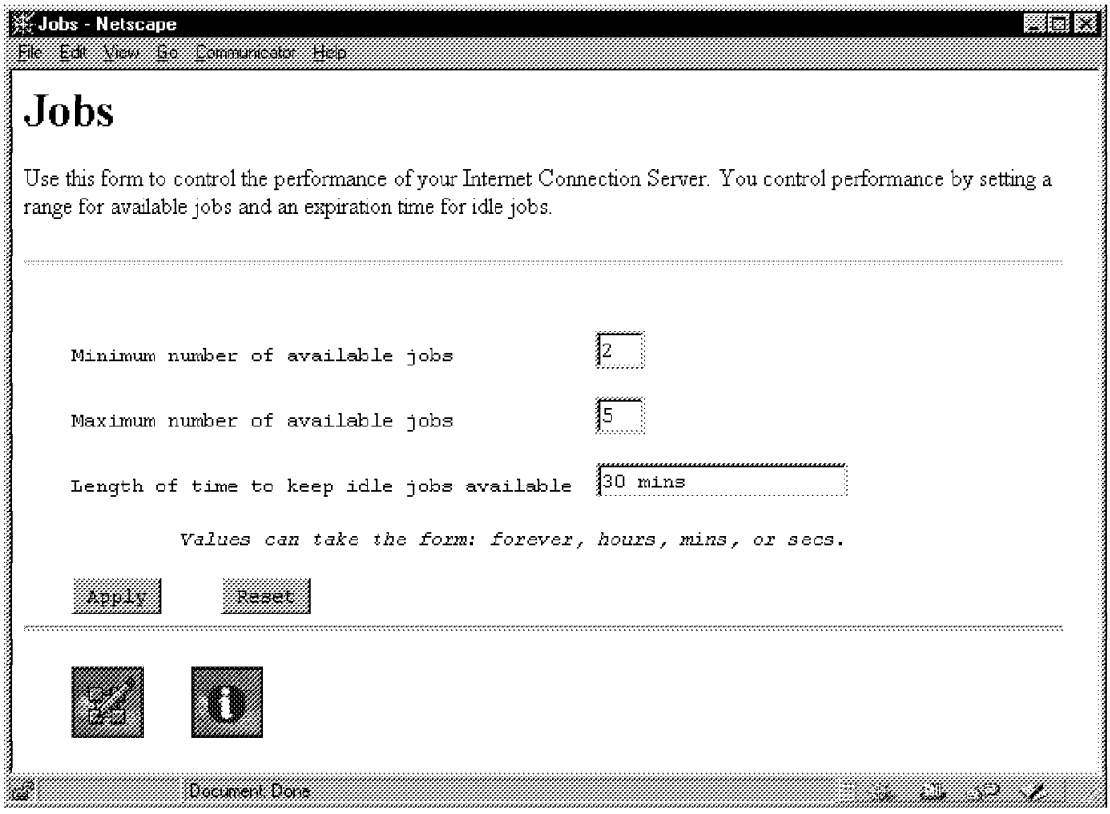


Figure 70. Jobs

To change the performance settings:

1. Fill in the following fields:
   - **Minimum number of available jobs -** the minimum number of jobs that you want the server to either be using or have available to use. The server does not close available jobs below this minimum even if the jobs are idle. Generally, the more requests the server receives, and the more power the machine has, the higher the value you should use for this directive. The value used can be overriden through instance parameters (see Figure 35 on page 56) or instance start-up values (see 6.11, “Start TCP/IP Server and End TCP/IP Server Commands” on page 102).
   - **Maximum number of available jobs** - the maximum number of jobs that you want to have active at one time. If the maximum is reached, the server holds new requests until another request finishes and jobs become available. Generally, the more requests the server receives, and the more power the machine has, the higher the value you should use for this directive. The value used can be overriden through instance parameters (see Figure 35 on page 56) or instance start-up values (see 6.11, “Start TCP/IP Server and End TCP/IP Server Commands” on page 102).
   - **Length of time to keep idle jobs available -** the amount of time the server should keep an idle job available. A job becomes idle after the last request to use it completes. If the number of jobs already available or active is greater than the value you specified for **Minimum number of**

available jobs, and the server does not use the job again within the specified time, the server closes the idle job. Specify a time using any combination of hours, mins (minutes), and secs (seconds). If you do not want the server to close any idle jobs, specify forever as the value. Some examples are:

30 secs
15 mins
20 mins
30 secs

2. Click **Apply** to update the server with the changes you made to the form, or click **Reset** to return to the values that were on the form before you made the changes.

```
                HTTP Configuration File

          MinActiveThreads       2
          MaxActiveThreads       5
          IdleThreadTimeout       30 mins
```

Figure 71. Timeouts Settings Added to HTTP Configuration File

Figure 71 shows the results of applying the settings in the form shown in Figure 70 on page 100.

6.11 Start TCP/IP Server and End TCP/IP Server Commands

In addition to being able to start and stop server instances from the ADMIN server browser interface (see Figure 31 on page 49), they can be started and stopped from a green screen using the STRTCPSVR and ENDTCPSVR commands.

To start a server instance:

```
STRTCPSVR SERVER(*HTTP) HTTPSVR(instance_name 'instance_startup_values')
```

Instance startup values: IBM recommends that you use these overrides with caution; they are intended only for special circumstances. Use the configuration and administration forms to specify start-up values. Instance start-up values specified on this parameter take precedence over configuration data in instance information and configurations.

-netccsid (nnn) Overrides the DefaultNetCCSID directive

-fsccsid (nnn) Overrides the default DefaultFsCCSID directive

-p (nnnn) Overrides the Port directive

-sslport (nnnn) Overrides the SSLPort directive

-r (configuration file) Overrides the configuration file for this instance of the server

-l (log-file-name) Overrides the access log file name

-newlog (log-file-name) Overrides the access log file name and sets the log format to common

-ddslog (log-file-name) Overrides the access log file name and sets the log format to DDS

-errlog (log-file-name) Overrides the error log file name

-minat (nn) Overrides the MinActiveThreads directive

-maxat (nn) Overrides the MaxActiveThreads directive

Example of specifying startup values for an HTTP instance:

```
STRTCPSVR SERVER(*HTTP) HTTPSVR(HTTP1 '-p 81 -sslport 443')
```

This command starts the server instance named http1 and specifies that the server instance should listen on port 81 for non-secure requests and on port 443 for secure requests. The ports defined here will override any previously defined ports to be used by this server instance.

To end a server instance:

```
ENDTCPSVR SERVER(*HTTP) HTTPSVR(instance_name)
```

Chapter 7. How to Get ICS for AS/400 Server Up and Running

In this chapter, we show the steps to get an ICS for AS/400 server up and running with minimal configuration. We follow these steps:

- Configure and start TCP/IP
- Create a welcome file
- Authorize the server to access the welcome file
- Create a new server instance and configuration
- Start the server
- Use a Web browser and request the home page

Before proceeding with the following steps, check that the TCP/IP Connectivity Utilities for AS/400 (5769-TC1) is installed. Internet Connection Server for AS/400 is part of 5769-TC1.

1. Get TCP/IP up and running on the AS/400 system. To check that a TCP/IP connection is configured on the AS/400 system, use the NETSTAT *IFC command.

```
                      Work with TCP/IP Interface Status
                                                              System:   SYSTEM01
Type options, press Enter.
  5=Display details   8=Display associated routes   9=Start   10=End
  12=Work with configuration status

     Internet         Network            Line      Interface
Opt  Address          Address          Description  Status
__   10.5.69.212      10.5.69.192      TRNLINE      Active
__   127.0.0.1        127.0.0.0        *LOOPBACK    Active
```

Figure 72. Verifying the TCP/IP Interface Status

   For more information about configuring the TCP/IP connection, see *TCP/IP Configuration and Reference*, SC41-5420.

2. Decide which file system will contain the documents and objects you will serve from the AS/400 HTTP server.

   In this example, we use the ′root′ file system.

3. Create a directory, folder, or library to store documents.

   To create a directory, we use the CRTDIR ′/TEST′ command.

4. Create a welcome file (this is an HTML document).

   We copy the ICS for AS/400 sample home page to our directory. This sample home page includes two gif files. We, therefore, have to copy these two gif files and the html page itself.

```
                              Copy Object (CPY)

 Type choices, press Enter.

 Object . . . . . > '/QIBM/ProdData/HTTP/Public/TC1/ICSS/HTML/gr
aphic4.gif'
 To directory . . . . . '/test/'
______
 To object  . . . . . 
______
 Symbolic link  . . . . . *NO           *NO, *YES
 To Code Page . . . . . *OBJ          1-32767, *OBJ, *CALC
 Data Format  . . . . . *BINARY       *BINARY, *TEXT
                                                                  Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 73. Using Copy Object to Copy a gif File

```
                              Copy Object (CPY)

 Type choices, press Enter.

 Object . . . . . > '/QIBM/ProdData/HTTP/Public/TC1/ICSS/HTML/sa
mpmast.gif'
 To directory . . . . . '/test/'
______
 To object  . . . . . 
______
 Symbolic link  . . . . . *NO           *NO, *YES
 To Code Page . . . . . *OBJ          1-32767, *OBJ, *CALC
 Data Format  . . . . . *BINARY       *BINARY, *TEXT
                                                                  Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 74. Using Copy Object to Copy a gif File

```
                              Copy Object (CPY)

 Type choices, press Enter.

 Object . . . . . > '/QIBM/ProdData/HTTP/Public/TC1/ICSS/HTML/We
lcome.html'
 To directory . . . . . '/test/'
______
 To object  . . . . . 
______
 Symbolic link  . . . . . *NO           *NO, *YES
 To Code Page . . . . . *OBJ          1-32767, *OBJ, *CALC
 Data Format  . . . . . *BINARY       *BINARY, *TEXT
                                                                  Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 75. Using Copy Object to Copy an HTML File

5. Authorize the server to access the welcome file.

Check that the user profile QTMHHTTP has sufficient authorities (at least *RX) to the objects in the directory. By default, this is the user profile that the server runs under. For example:

```
dspaut '/test'
dspaut '/test/Welcome.html'
```

```
                              Display Authority

Object . . . . . :   /test
Owner  . . . . . :   HEIKKI
Primary group  . . . . . :   *NONE
Authorization list . . . . . :   *NONE


              Data     --Object Authorities--
User        Authority  Exist  Mgt  Alter  Ref
*PUBLIC     *RWX         X     X     X     X
HEIKKI      *RWX         X     X     X     X

                                                                          Bottom
Press Enter to continue.

F3=Exit   F11=Display detail data authorities   F12=Cancel   F17=Top
F18=Bottom
(C) COPYRIGHT IBM CORP. 1980, 1997.
```

Figure 76. Verifying the Object Authorization

```
                              Display Authority

Object . . . . . :   /test/Welcome.html
Owner  . . . . . :   HEIKKI
Primary group  . . . . . :   *NONE
Authorization list . . . . . :   *NONE


              Data     --Object Authorities--
User        Authority  Exist  Mgt  Alter  Ref
*PUBLIC     *RX
HEIKKI      *RWX         X     X     X     X
QSYS        *RWX         X     X     X     X
                                                                          Bottom
Press Enter to continue.

F3=Exit   F11=Display detail data authorities   F12=Cancel   F17=Top
F18=Bottom
(C) COPYRIGHT IBM CORP. 1980, 1997.
```

Figure 77. Verifying the Object Authorization

If the server user profile or *PUBLIC does not have sufficient authority, use the CHGAUT command to grant authority. For example:

```
CHGAUT OBJ('/test/Welcome.html') USER(QTMHHTTP) DTAAUT(*RX)
```

6. Start the Administration Server:

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

7. Go to the AS/400 Tasks Page:

   Point your browser to http://yourserver:2001/.

Enter your AS/400 userid and password.

Note: If you are not familiar with the AS/400 Tasks page browser interface, see Chapter 5, “Web Browser Configuration Interface” on page 33.

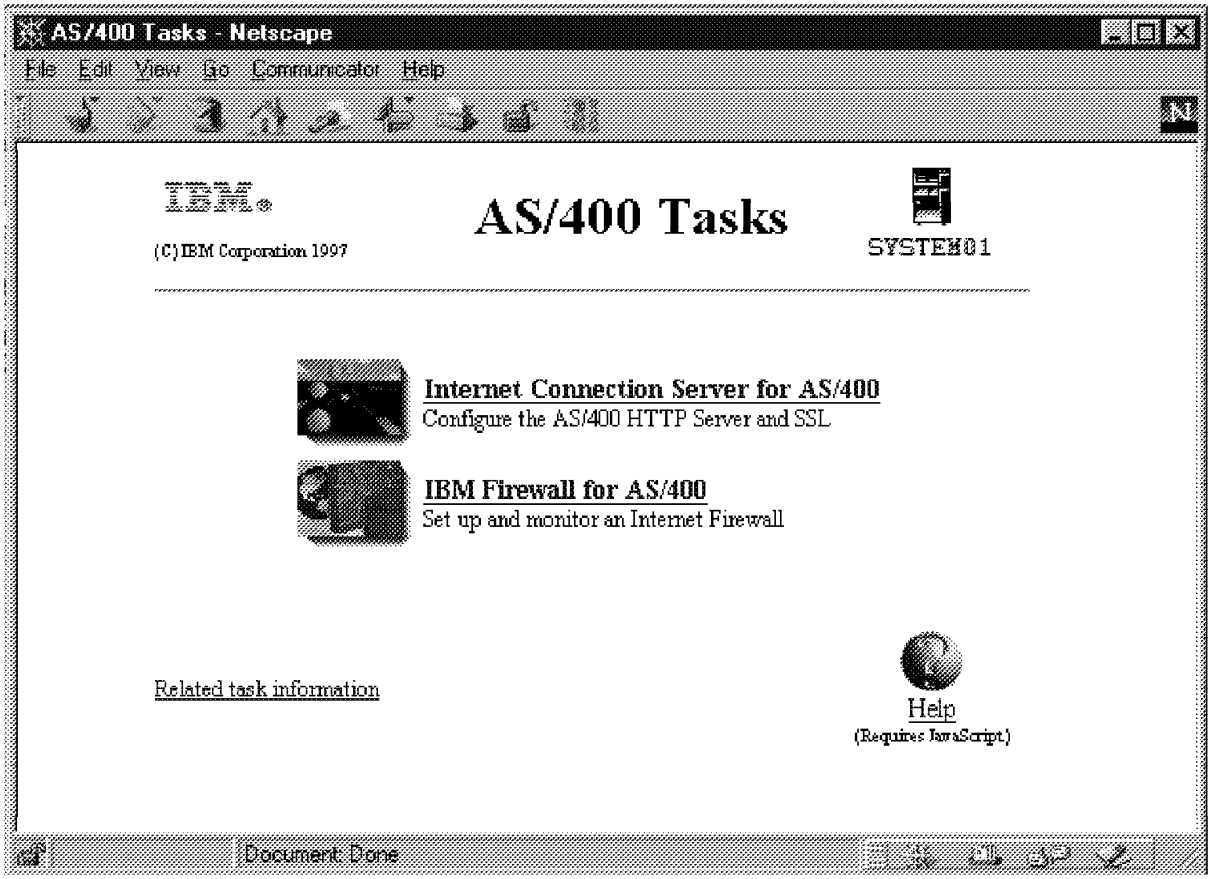


Figure 78. AS/400 Tasks Page

8. Create an new server instance:

   Go to the **General Configuration and Administration** page.

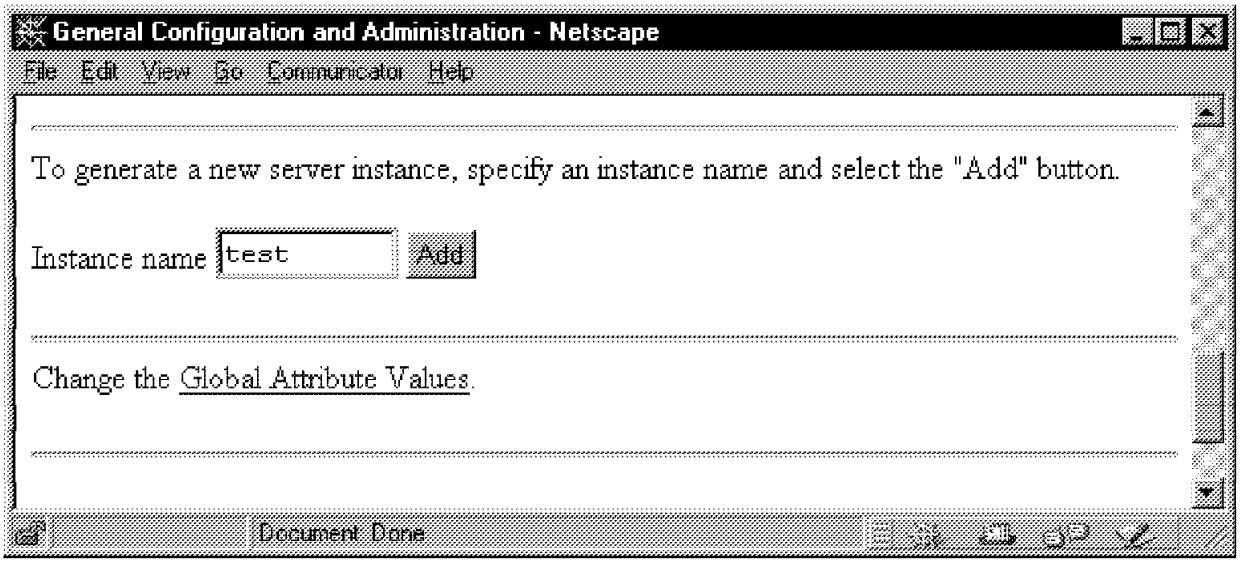


Figure 79. Adding a New Server Instance

Enter the new instance name and press **Add** to create the new server instance. Wait for the confirmation page and press the **Configuration Page** button to return to the General Configuration and Administration page.

9. Create a new server configuration for the server instance.

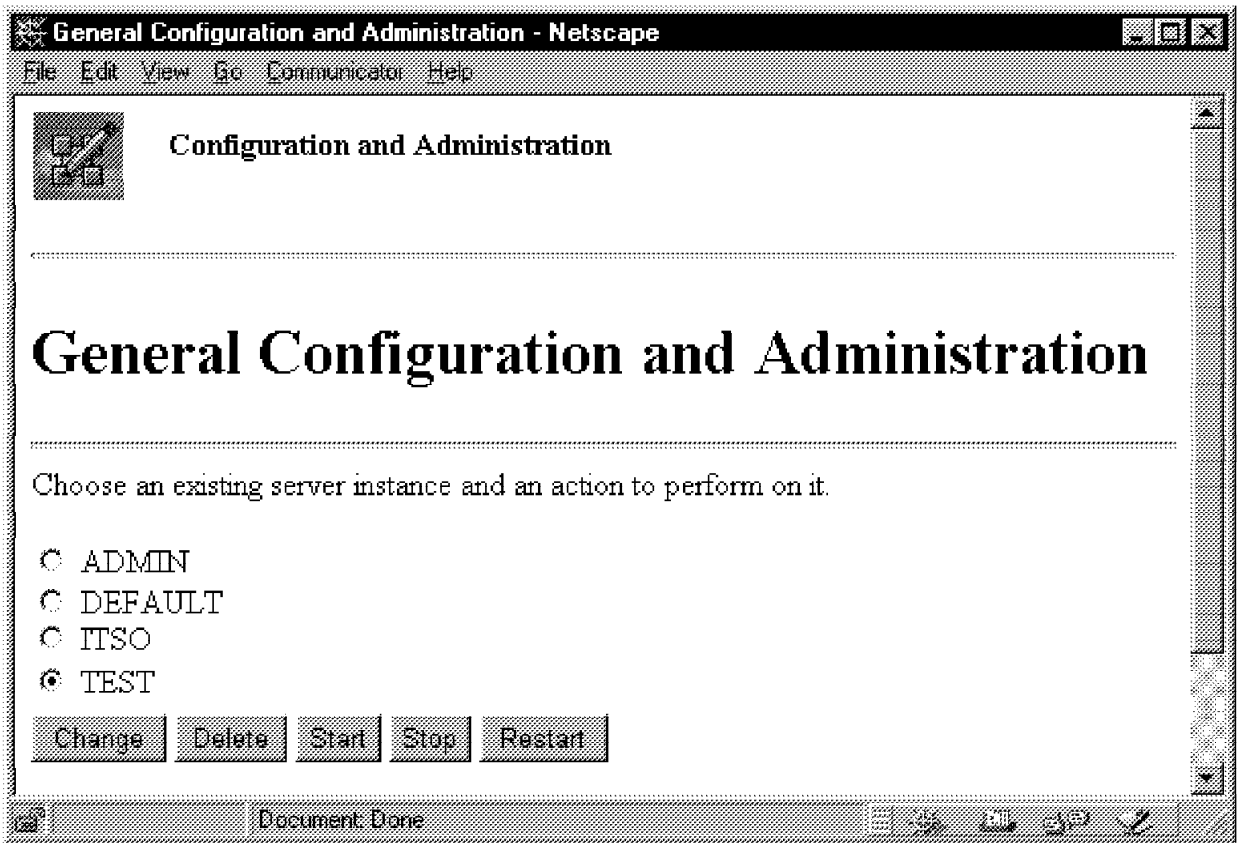


Figure 80. Selecting the New Server Instance

Select the new server instance and press **Change**.

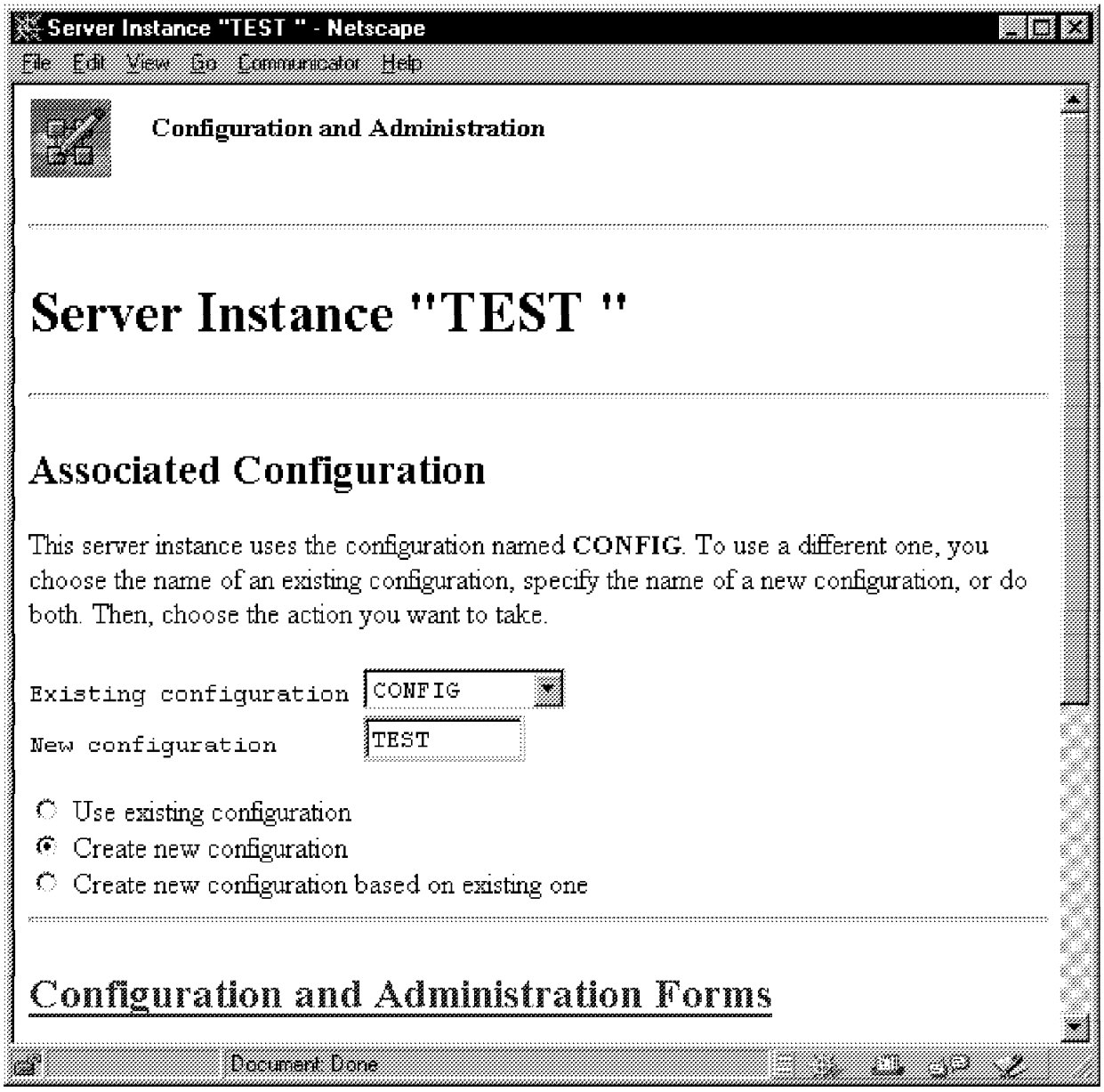


Figure 81. Creating the New Configuration

Enter the new configuration name, select to create a new configuration, and press **Apply**. Wait for the confirmation page and press the **Configuration Page** button to return to the General Configuration and Administration page.

10. From the General Configuration and Administration page, select the new configuration and press the **Change** button to be presented with the Server Instance form.

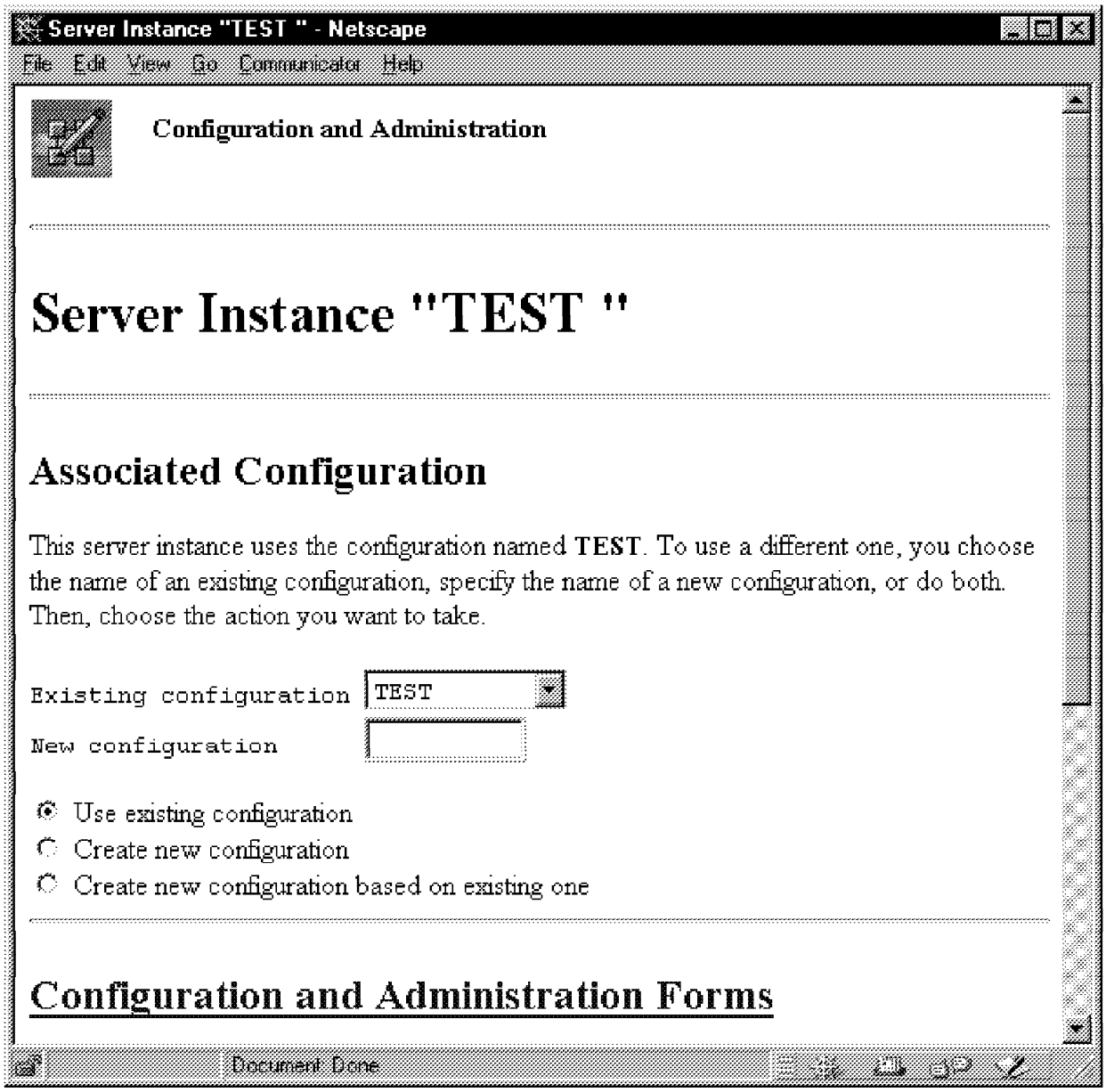


Figure 82. Selecting the New Configuration

Select the **Configuration and Administration Forms** page.

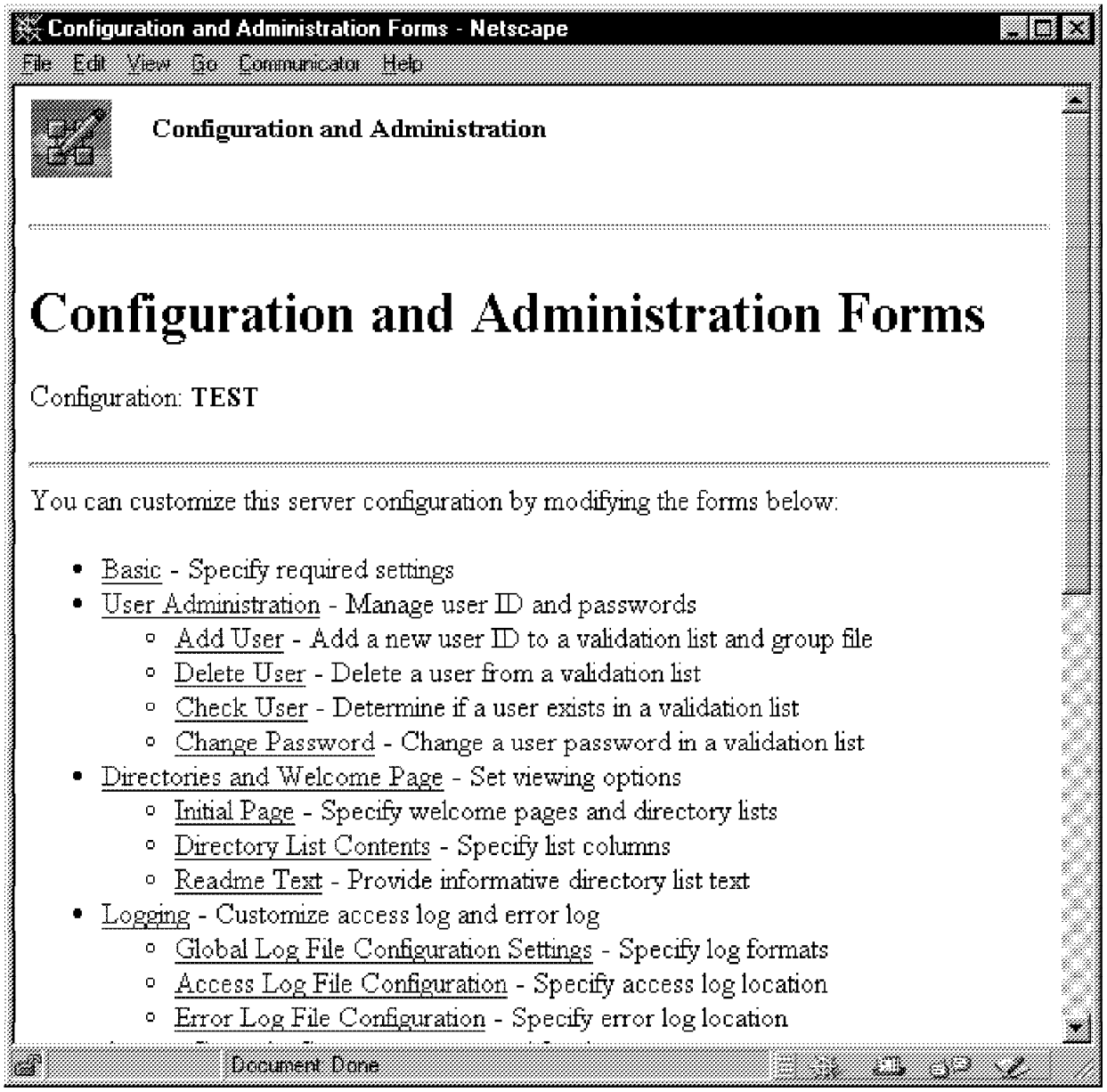


Figure 83. Configuration and Administration Forms

11. Define Basic settings:

    Select **Basic**.

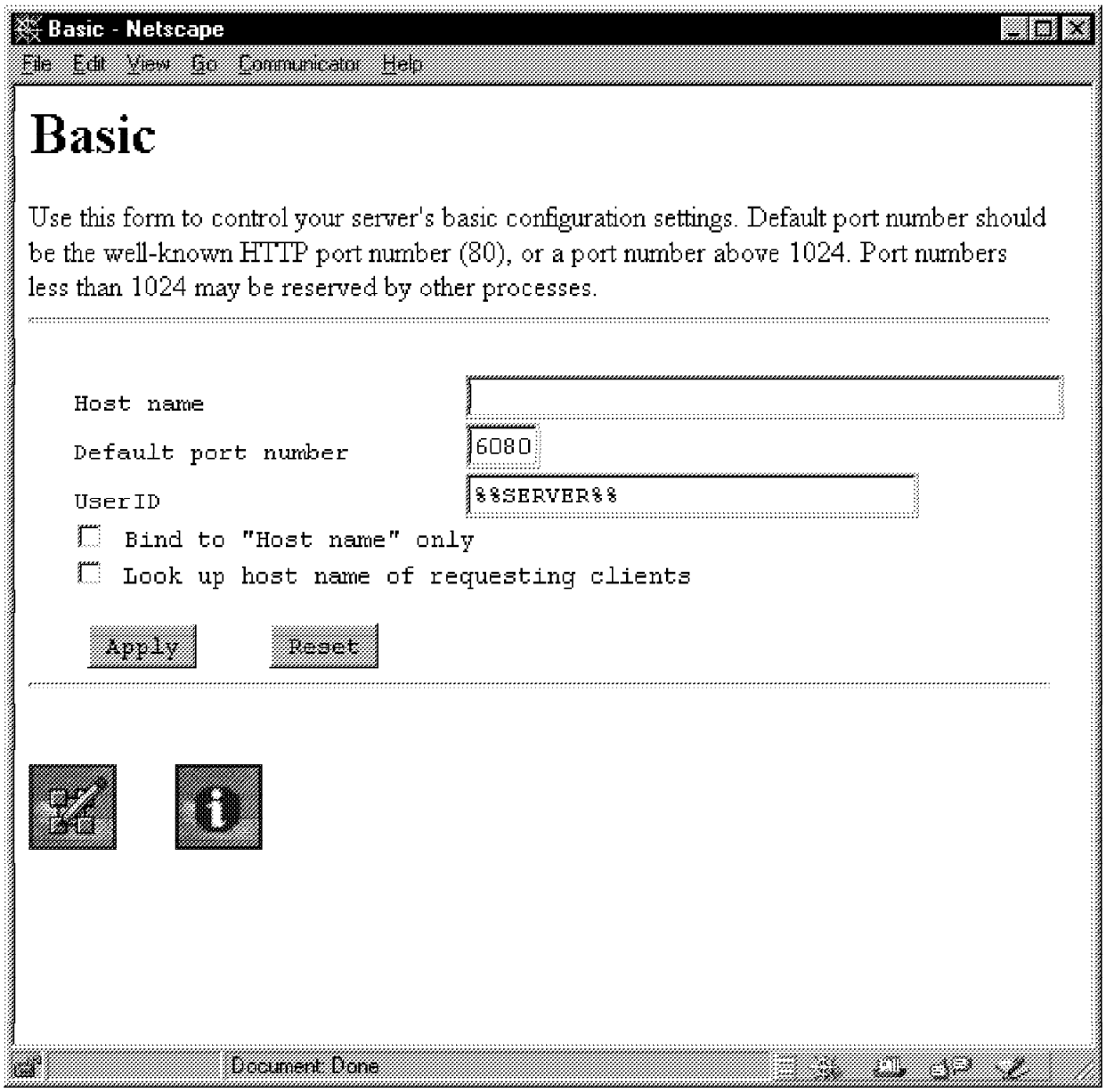


Figure 84. Defining the Basic Server Settings

Select a host name and, if required, a port number for the server instance. We selected a non-standard port, but you do not have to.

Note: Look up host name of requesting clients is selected by default.

Press the **Apply** button. Wait for the confirmation page and press the **Configuration Page** button.

Figure 85. Confirmation Page

12. Change directory list viewing.

    Select **Initial Page** under Directories and Welcome Page and scroll down to Directory List Viewing.

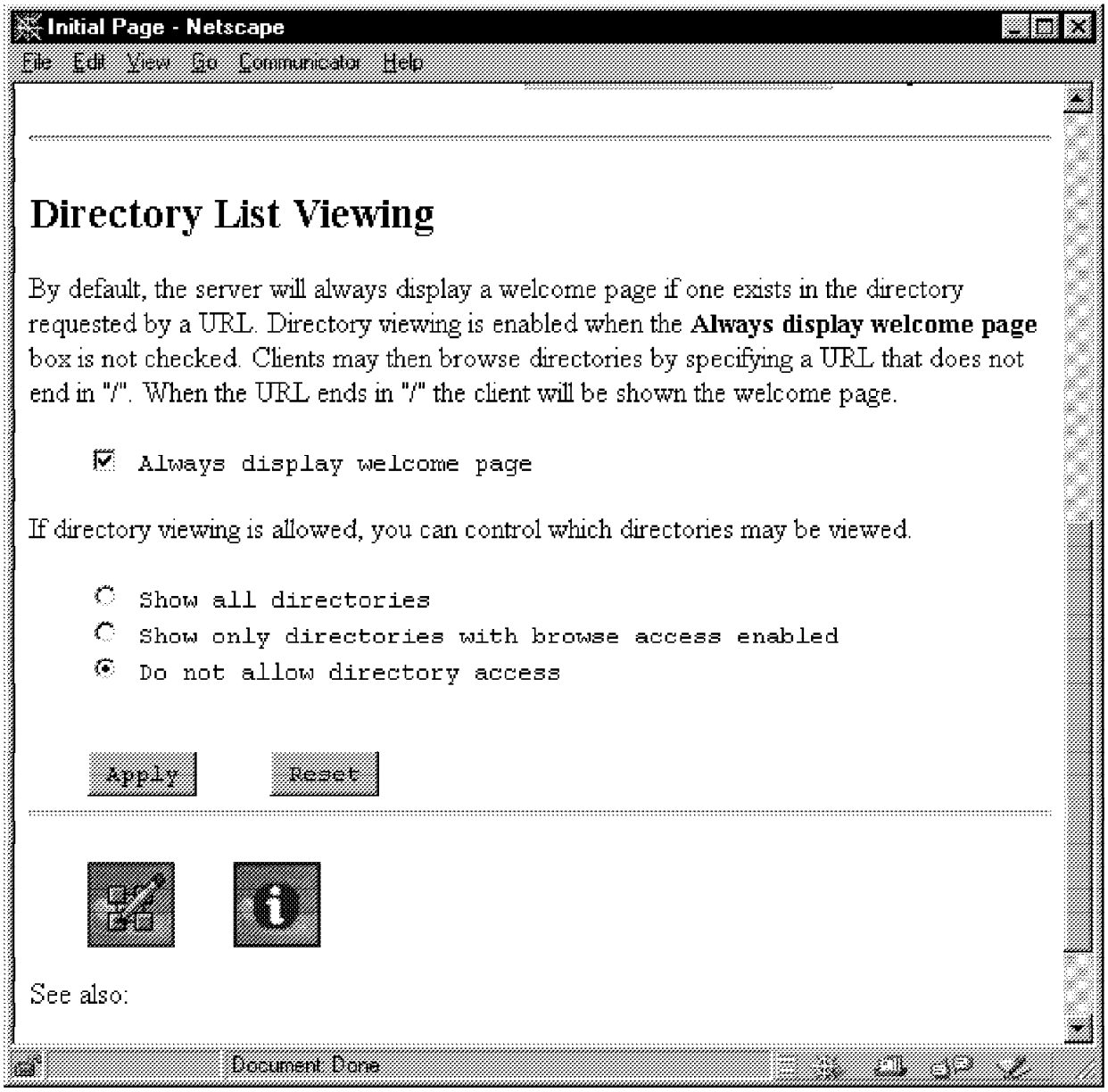


Figure 86. Directory List Viewing

We selected to not allow directory access. **Show all directories** is selected by default. Press the **Apply** button. Wait for the confirmation page and press the **Configuration Page** button.

Figure 87. Confirmation Page

13. Add a Pass statement for the Welcome page:

    Select **Request Routing** under Resource Mapping.

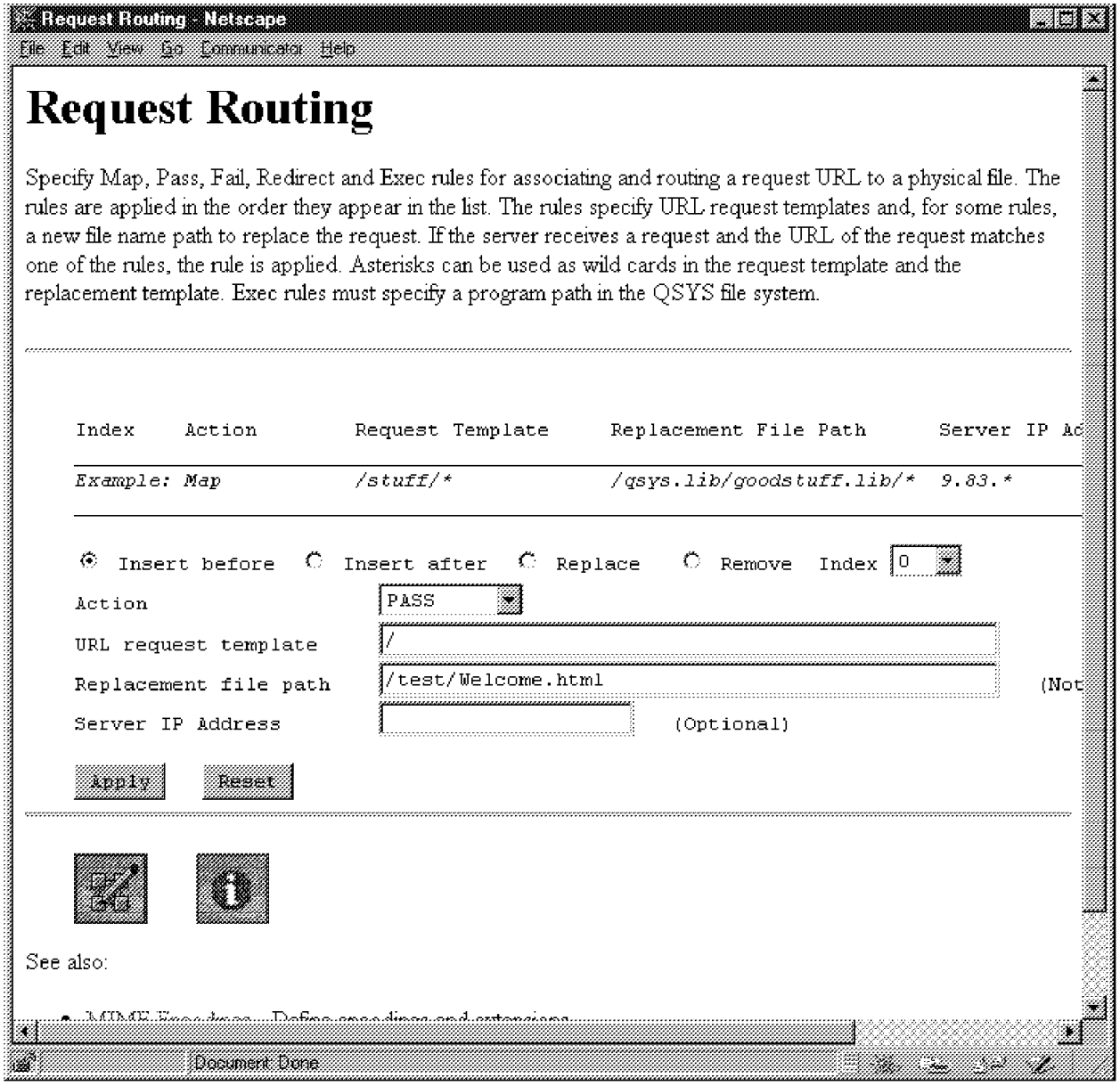


Figure 88. Request Routing

The pass statement entered in Figure 88 serves the welcome page. Press the **Apply** button. Wait for the confirmation page and press the **Configuration Page** button.

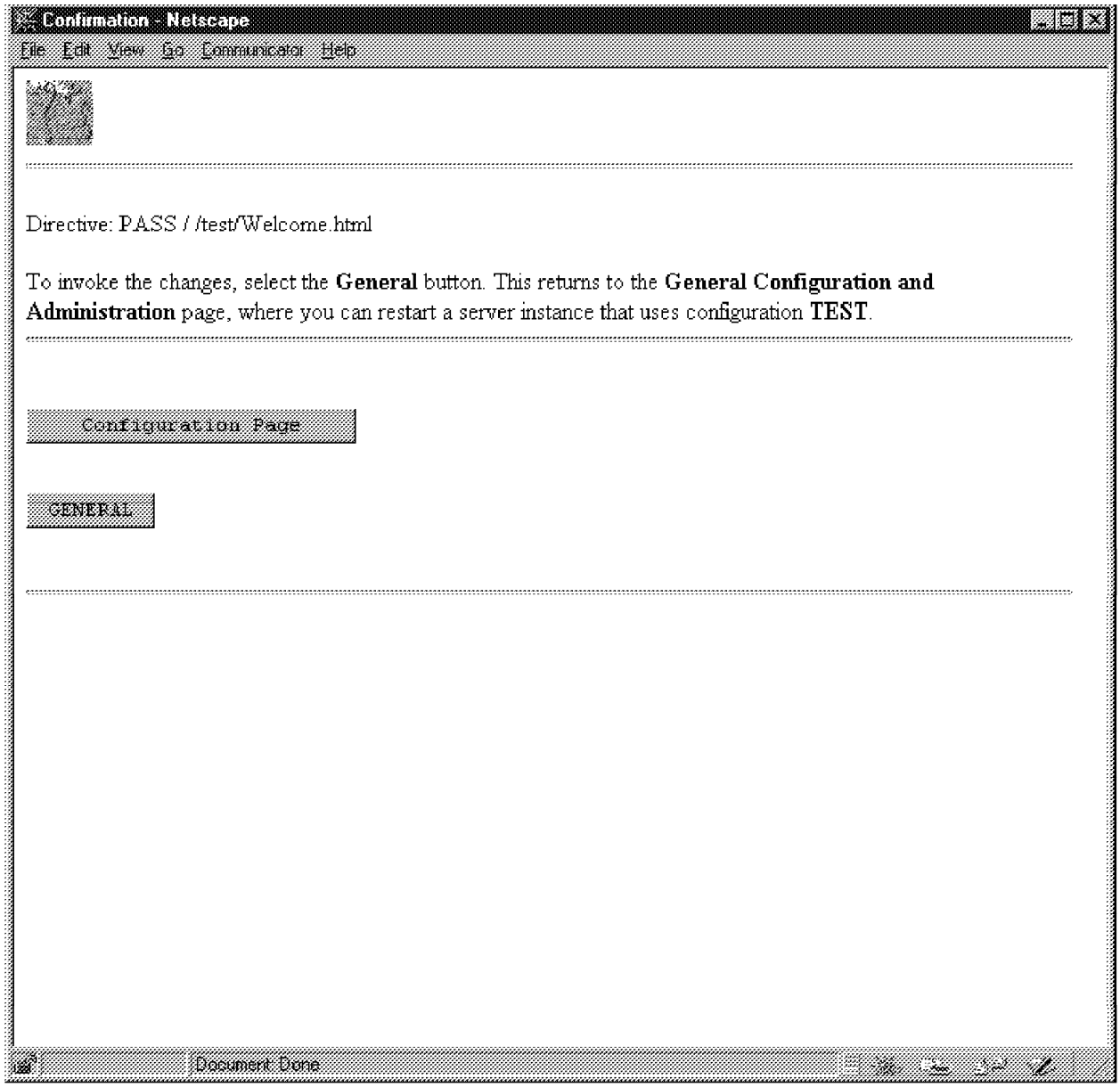


Figure 89. Confirmation Page

14. Add a Pass statement for the Welcome page GIF files:

    Select **Request Routing** under Resource Mapping.

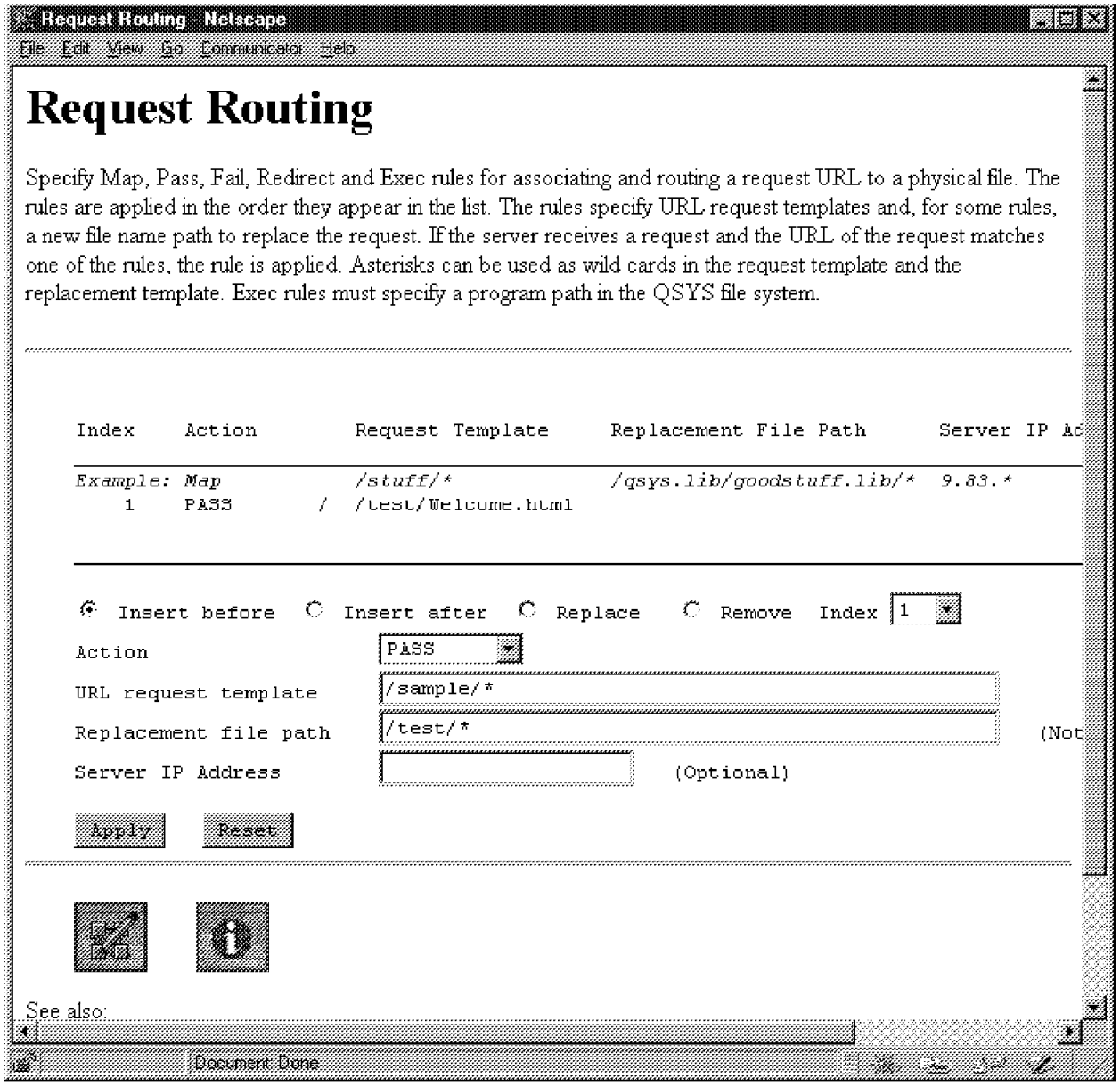


Figure 90. Request Routing

The pass statement entered in Figure 90 maps requests for the GIFs to the test directory. The GIFs have a relative URL of /sample in the welcome page. Press the **Apply** button. Wait for the confirmation page and press the **General** button.

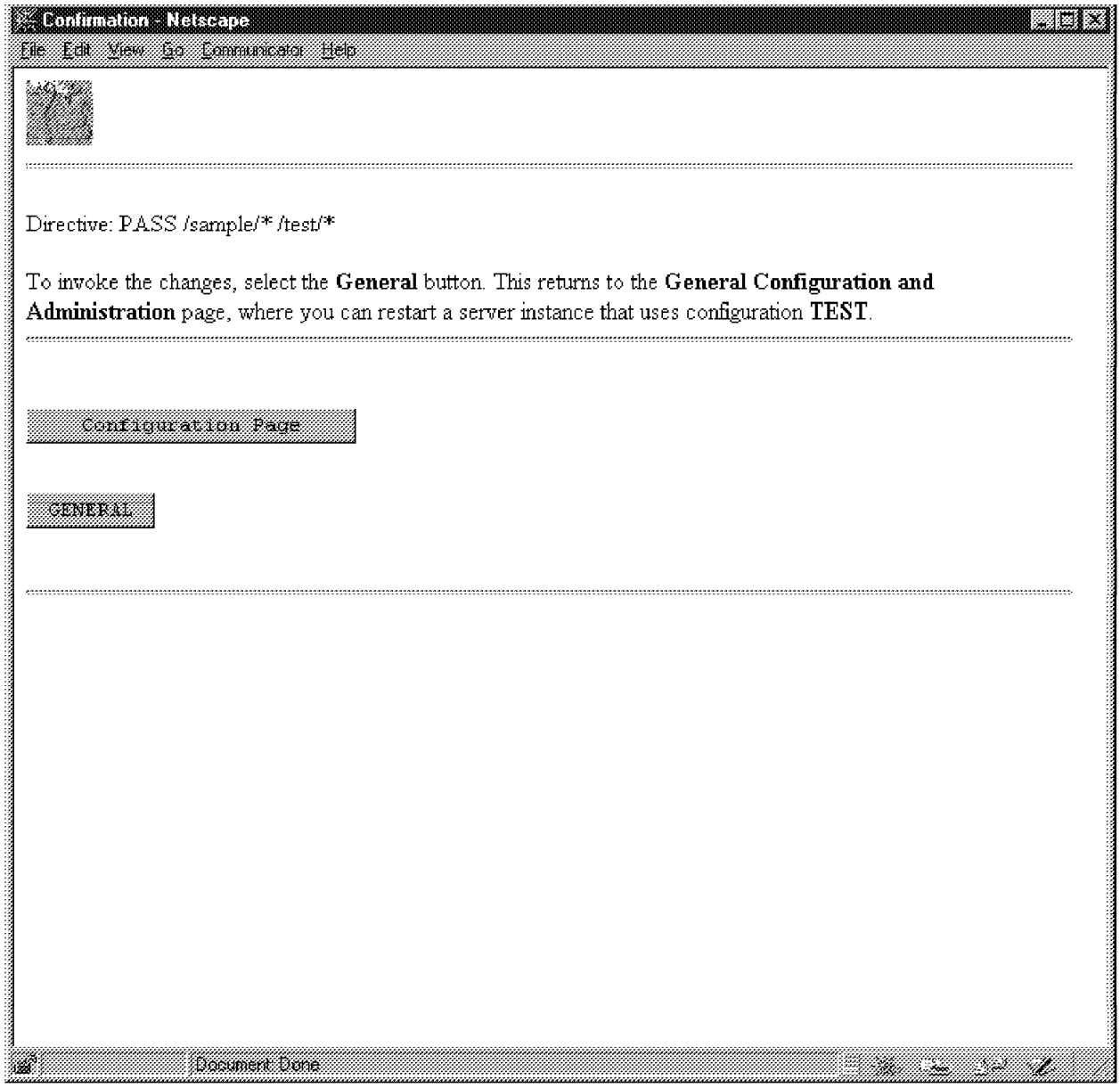


Figure 91. Confirmation Page

15. Start the server:

    Select the new server instance and press the **Start** button.

16. Test the server:

    You can check that the server is running. For example:

    ```
    WRKACTJOB JOB(TEST)
    ```

```
                              Work with Active Jobs                  SYSTEM01
                                                                  11/25/97  16:37:26
CPU %:     .0     Elapsed time:   00:00:00     Active jobs:   260
Opt  Subsystem/Job  User        Type  CPU %  Function        Status
__     TEST         QTMHHTTP    BCH      .0  PGM-QTMHHTTP     TIMW
__     TEST         QTMHHTTP    BCI      .0                   DEQW
__     TEST         QTMHHTTP    BCI      .0                   DEQW
__     TEST         QTMHHTTP    BCI      .0                   DEQW
__     TEST         QTMHHTTP    BCI      .0                   SELW
__     TEST         QTMHHTTP    BCI      .0                   SELW
```

Figure 92. Using WRKACTJOB to Verify Server is Running

And you can check that the port you defined is active. For example: NETSTAT *CNN then press F14.

```
                        Work with TCP/IP Connection Status
                                                          System:   SYSTEM01
Local internet address  . . . . . . . . . . . :   *ALL

Type options, press Enter.
  4=End   5=Display details

     Remote           Remote     Local
Opt  Address          Port       Port       Idle Time  State
 _   *                *          1502       021:50:09  *UDP
 _   *                *          1503       097:57:21  *UDP
 _   *                *          as-admi >  000:16:02  Listen
 _   *                *          2049       021:50:09  *UDP
 _   *                *          2080       003:37:43  Listen
 _   *                *          wsg        001:22:13  Listen
 _   *                *          5880       026:30:50  Listen
 _   *                *          5881       023:31:37  Listen
 _   *                *          6010       099:09:45  Listen
 _   *                *          6012       343:40:54  Listen
 _   *                *          6080       000:11:55  Listen
                                                                    More...
F5=Refresh   F11=Display byte counts   F13=Sort by column
F14=Display port numbers   F22=Display entire field   F24=More keys
```

Figure 93. Using NETSTAT to Verify the Server is Listening for Requests

Look to verify that a server is listening on either the default HTTP port (80) or the specific port entered (6080 in our case).

As a final verification step, we access the new server.

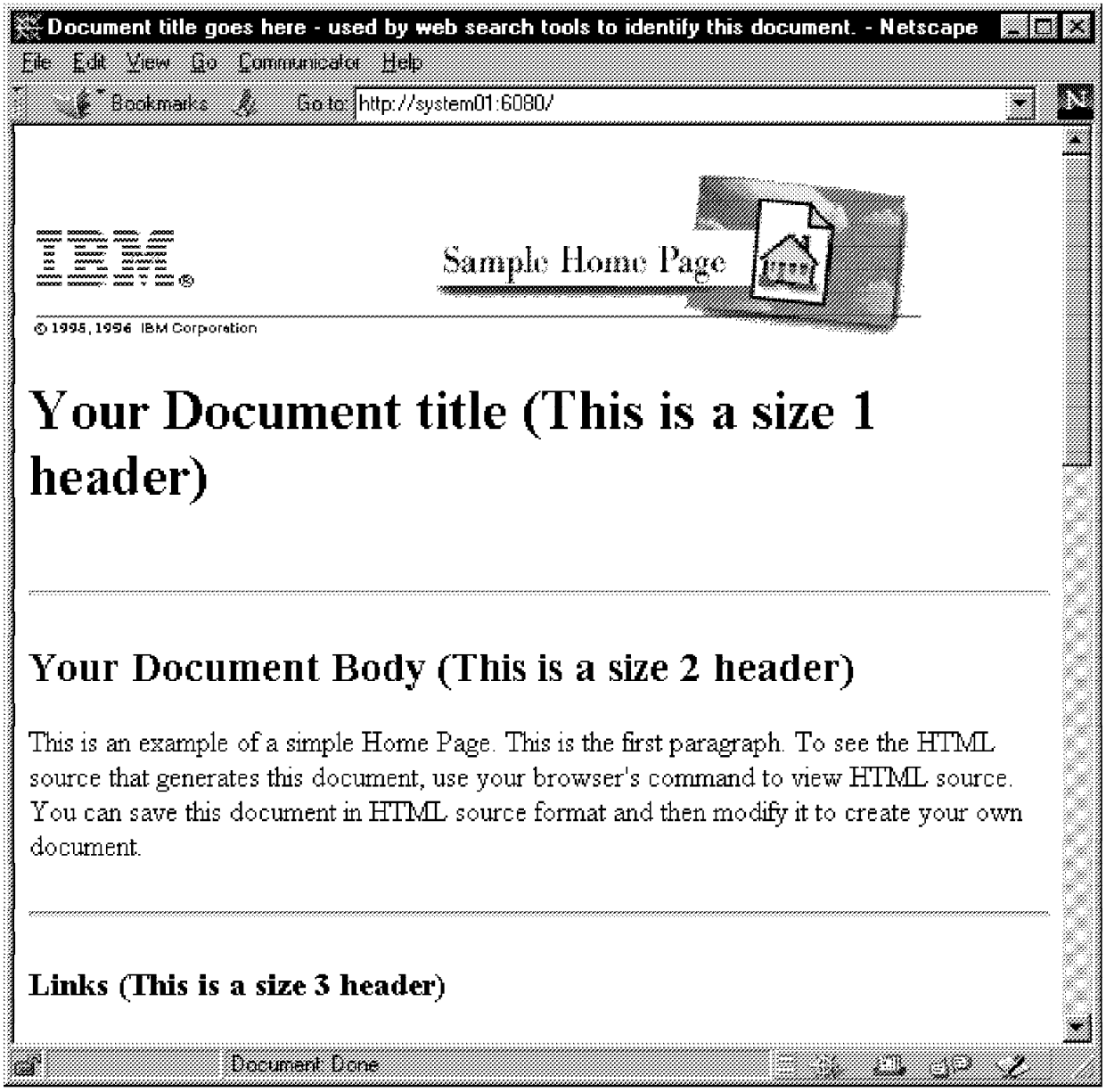


Figure 94. Server Welcome Page

It works!

Chapter 8. Protecting Server Resources

In this chapter, we look at access control as a method of protecting server resources. Chapter 6, “Basic Internet Connection Server Configuration” on page 43 shows how we can use Methods and Request Routing to limit access to server resources. These methods, however, are global in that they restrict access for all users. Access control provides a method of restricting access to server resources to individual users or groups of users. We do this through the use of protection directives.

Through the administration server, we can configure a server such that documents are protected using userid/password protection, address (IP address) template protection, or a combination of both of these. This allows an ICS for AS/400 or ICSS for AS/400 server to serve files and documents in some directories with no restrictions but to restrict access to files and directories in other directories to:

- Specific userids (with passwords)
- Groups of users (with userids and passwords)
- Specific IP addresses
- Ranges of IP addresses
- A combination of the preceding restrictions

Simply put, you can password-protect Web pages.

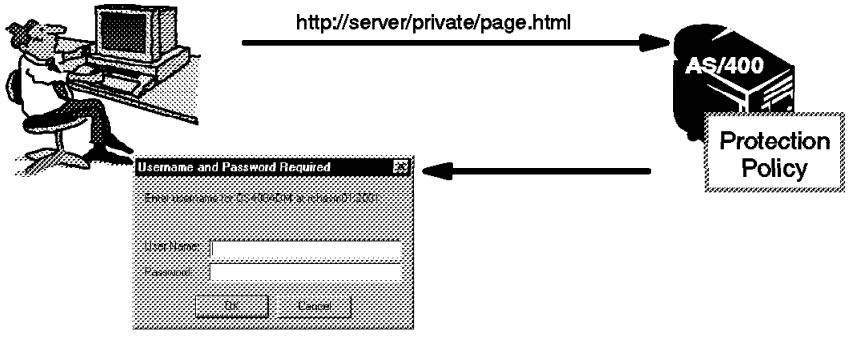


Figure 95. Password Protected Web Pages

When used with the AS/400 system′s Secure Sockets Layer (SSL) capability discussed in Chapter 9, “Establishing a Secure Connection” on page 159, the userid/password combination is encrypted before it is sent to the AS/400 system, thus making it virtually impossible for anyone such as a hacker to determine.

Access control is extremely useful if certain users of a Web site have different requirements than others. Let′s say we want all users to have access to the public domain files and, hence, we do not implement access control on the directories that contain these files. We do, however, want to restrict access to other documents and files and so we implement access control on the directories that contain these files. In Figure 96 on page 122, you can see that our company home page has several document links. Some of these document links such as the mission statement and site maps are public domain. Others such as administration functions and prices for distributors are protected by access control; we have indicated this by placing an image of a lock next to those links.

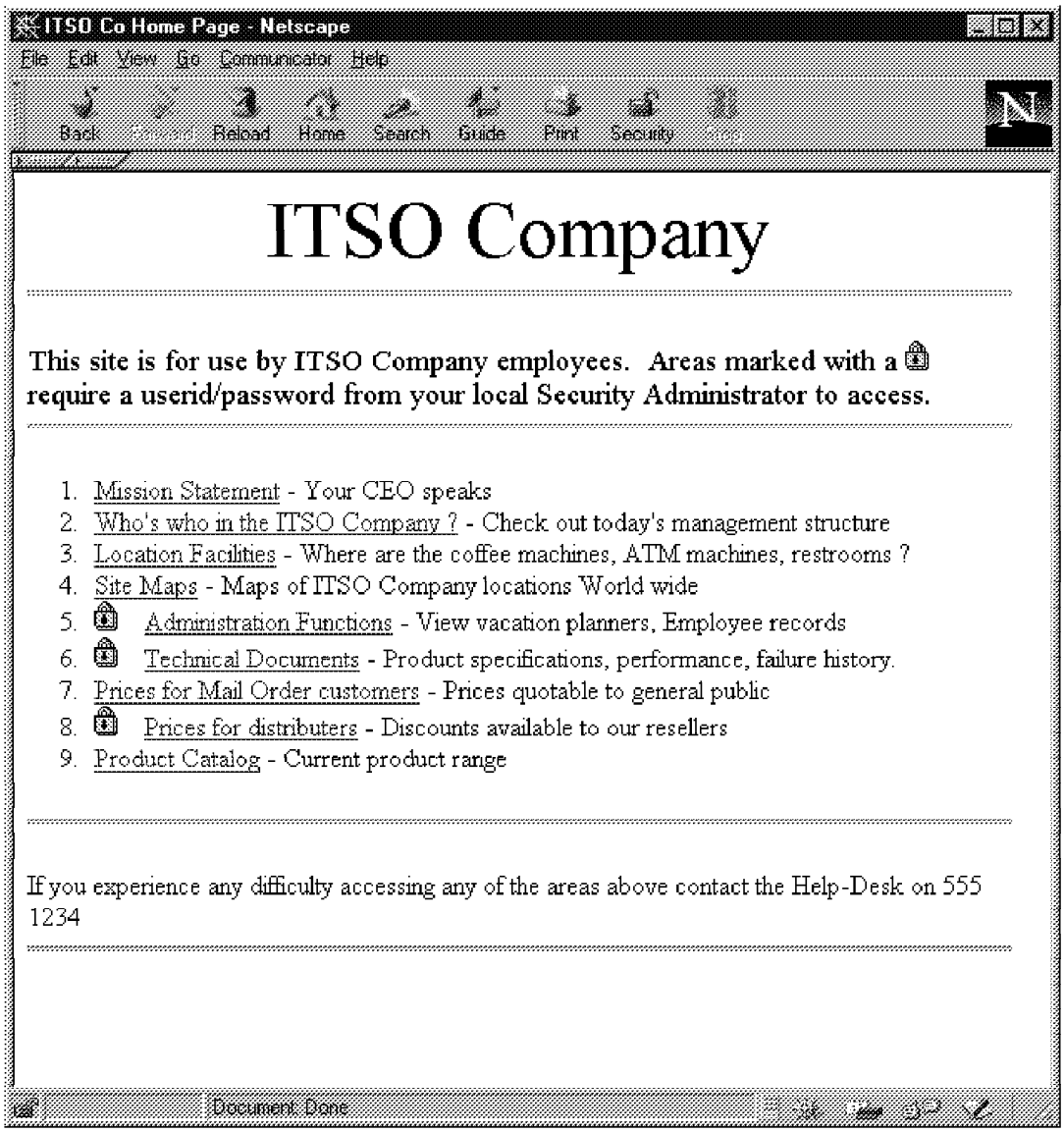


Figure 96. Web Page with Access Control Implemented

In this example, when a user selects a link where the directory or file is protected by access control, they receive a userid/password form to complete (see Figure 97 on page 123). The combination of userid and password entered must be correct for access to the protected directory to be granted.

Figure 97. Access Control Prompting for Userid and Password

Figure 98 shows the result of entering a userid/password combination that is not authorized to a protected directory.

Figure 98. Access Control Rejects Userid/Password Combination

It is also possible to use IP address template protection, either on its own or in combination with userid/password protection, to protect a directory or document. When implemented, Figure 99 on page 124 shows the result of a request from a user whose IP address is not authorized to a particular directory or file. This can be especially useful when we want to prevent access from any IP address outside of the corporate network (intranet).

When using address template protection only, the userid/password pop-up is not displayed. The request is rejected on the basis of the IP address alone.

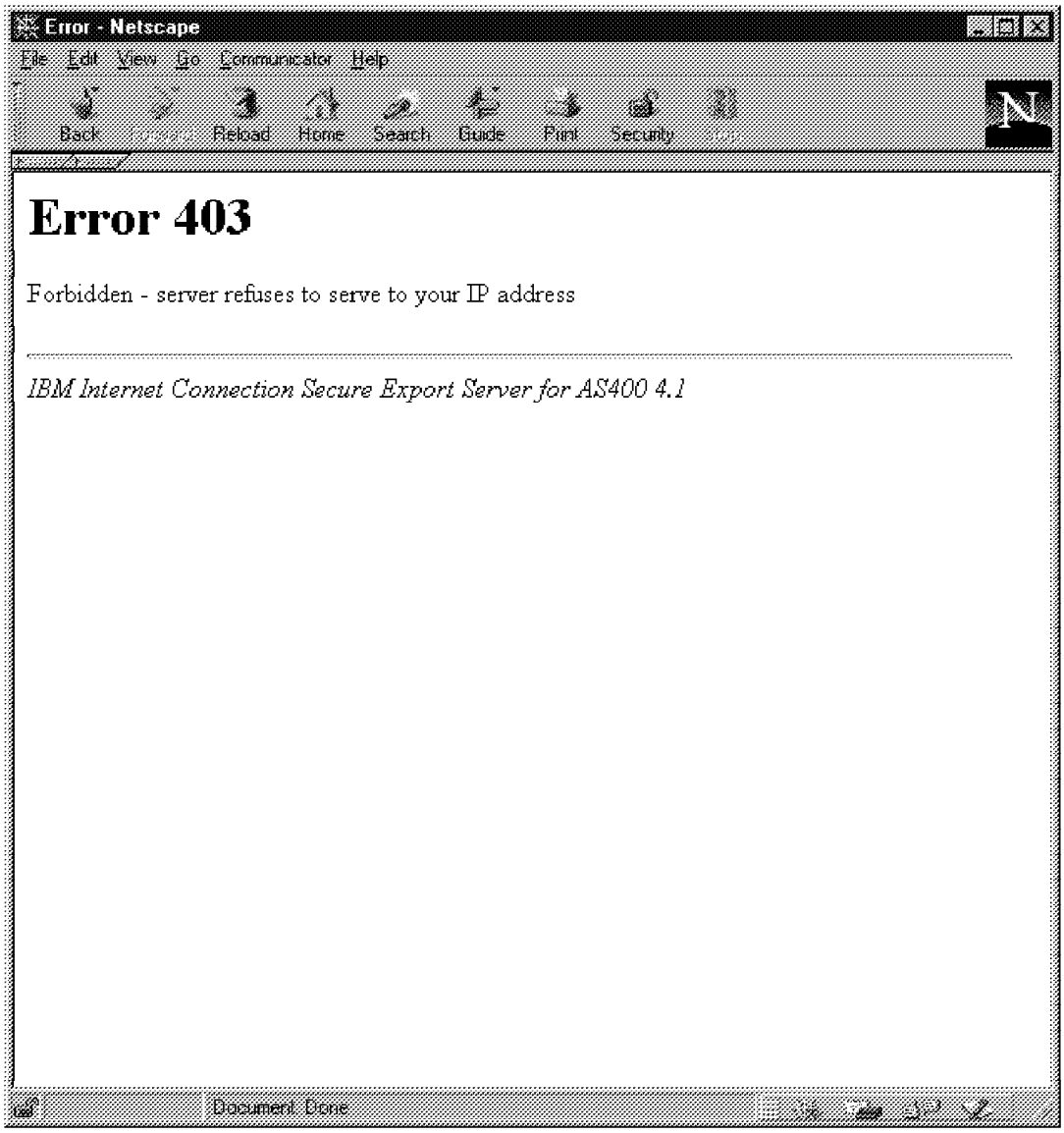


Figure 99. Access Control Rejects IP Address of Requester

Access control, when used in conjunction with other technologies such as Secure Sockets Layer (SSL) and the implementation of a firewall, provides for the implementation of a secure Web site.

8.1 Access Control Example

The implementation of Access Control typically involves two steps:

- User Administration: Adding users to a validation list and group file to allow access control to determine correct userid/password combinations of authorized users.
- Access Control: The implementation of protection directives to protect directory contents.

We shall look at each of these but first let's look at a simple example to introduce the concept of restricting access to directories based on user name, group name, and IP address.

In the example in Figure 100 on page 125, we have an Internet Connection Server for AS/400 on a corporate intranet. We are not concerned about external Internet hackers as the network is either not connected to the Internet or separated from it by means of a Firewall. We prefer, however, that the technical personnel are not able to view documents intended for the administration personnel and vice versa. By using userids/passwords and IP addresses, we can provide the following protection:

- Allow user SALLY to access documents in the Admin directory if she provides the correct userid/password and the IP address of the PC she makes the request from is x.x.x.x.
- Allow users in the group called TECHGRP to access documents in the Technical directory if they provide a valid userid/password and the request comes from IP address y.y.*.* (where * = any value).

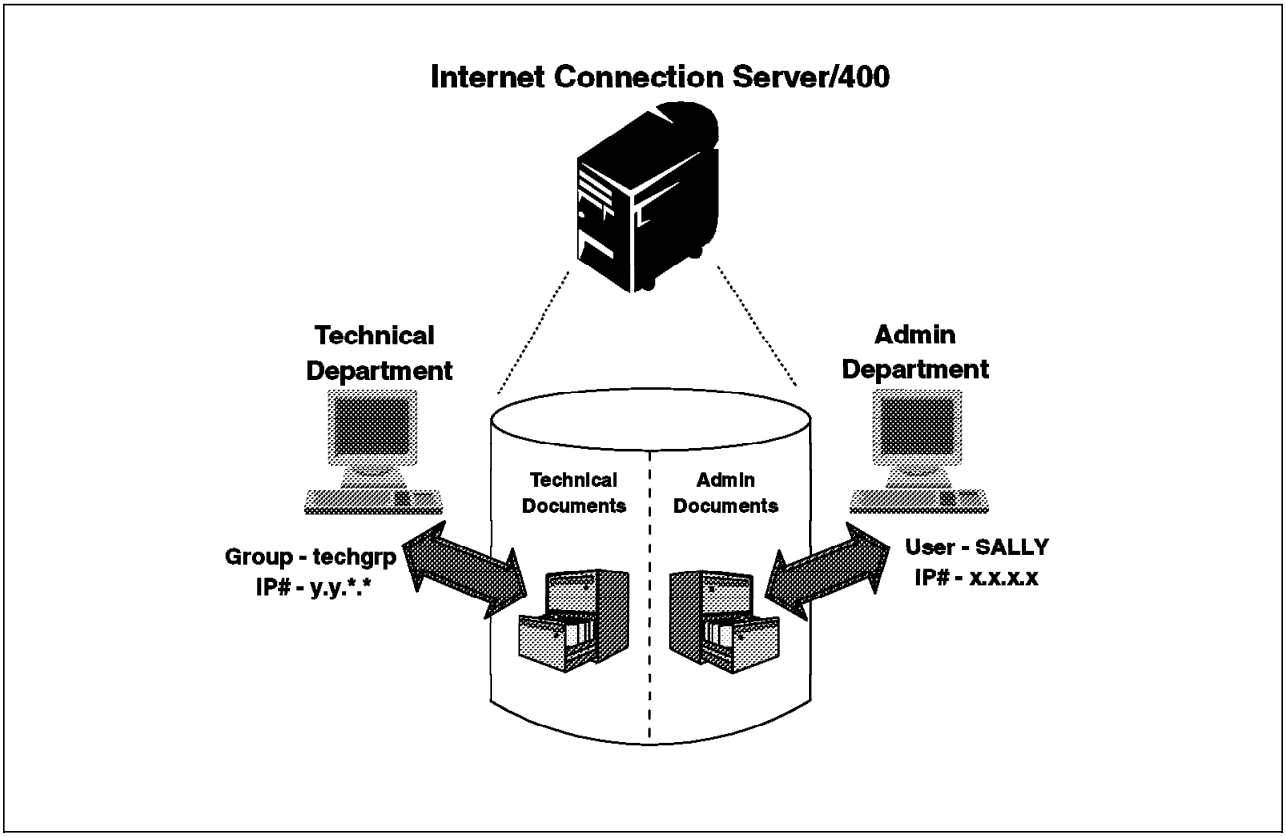


Figure 100. Restricting Document Access to Specific Users

In this example, we have implemented both userid/password protection and IP address template protection; this does not have to be the case. We can use just userid/password protection or just IP address template protection.

Note

To use address template protection, the client's IP address must be visible to the server. This is unlikely to be the case if the request is through the Internet where the client's IP address is likely to be hidden behind a Firewall. This means that implementing IP address template protection may only be of practical use on an intranet.

When using userid/password protection, we can either use Web server unique userids/passwords where these userids/passwords cannot be used to logon to the AS/400 system or we can use AS/400 userids/passwords. When using Web server unique userids and passwords, these userids/passwords are defined in a validation list.

8.2 Implementing Access Control

Protecting a server's resources involves the following areas:

User Administration: Where we manipulate userids and passwords contained within validation lists and group files that are used to access protected directories and files. User administration is only required if we are to use Web server unique userids/passwords rather than AS/400 userids/passwords.

Access Control: This is used to define which users can access which directories and files. Access control uses user profiles, either defined in User Administration or AS/400 user profiles, IP address templates, or a combination of these to determine access rights. Access control allows us to add rules (known as directives) that implement such control over our Web resources.

8.3 User Administration

User Administration is where we configure users, passwords, validation lists, group files, and groups. These configuration steps are required if we are to use Web server unique userids and passwords for access control.

First let's look at some of the terms we use:

User Name
The user id. User names are placed in a validation list and are not related to AS/400 user profiles. They are used to check the authority of people to server resources who do not have the requirement for an AS/400 user profile. This has the advantage that the user name is of no use at an AS/400 Sign On display.

Password The password associated with the user name. The password is stored with the user name in a validation list. Passwords in a validation list are not related to AS/400 user profiles.

Validation List
A validation list is a new AS/400 object of type *VLDL that stores user names and passwords for use in access control. Validation lists reside in AS/400 libraries.

Groups A group is a collection of users who require common access control to a directory or file. This might, for example, be people in the same department.

Group File
A group file holds information about which users belong to which groups. Group files reside in directories in the root file system.

Figure 101 on page 127 shows the relationship between these objects.

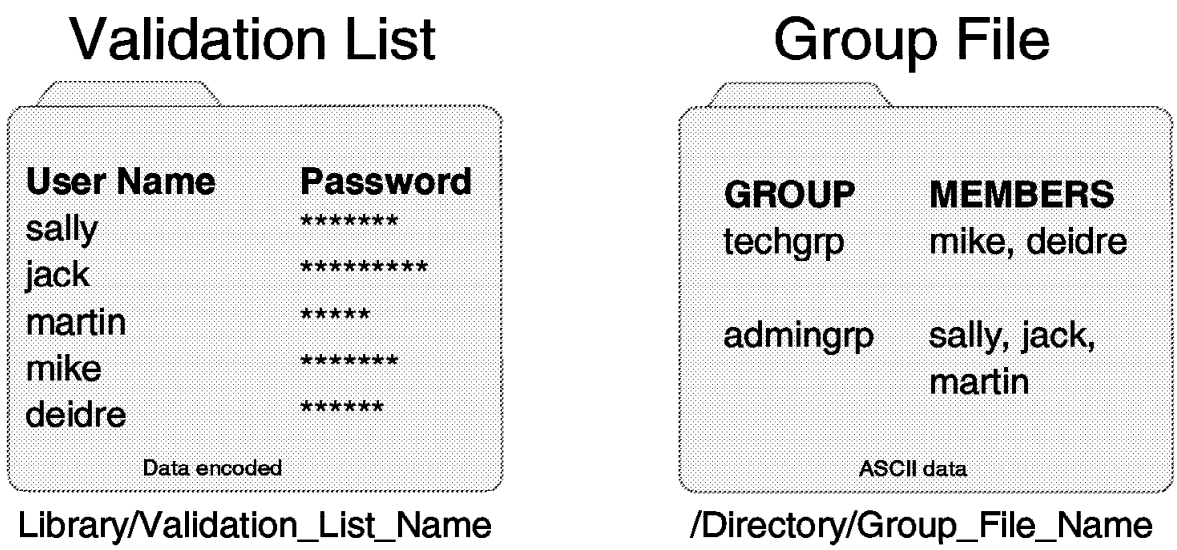


Figure 101. Validation List and Group File

Now that you understand the important terms, consider the following group of people in the ITSO company:

Table 5. ITSO Company Employees

User Name	Group
Sally	admingrp
Jack	admingrp
Martin	admingrp
Mike	techgrp
Deidre	techgrp

You can see in Table 5 that we have logically grouped the administration personnel (Sally, Jack, and Martin) into a group we call admingrp. Think of them as a group of workers in the same office or department who need access to administration files on the company server. Mike and Deidre work in a technical support department. They need access to technical documents on the company Web sever. We decided to put them into a group called techgrp. In each case, we prefer that each group is prevented from accessing documents intended for the other group. None of these users have existing OS/400 user profiles so we use validation lists and group files to perform the necessary authority checking.

8.3.1 Adding a User

Now that we have a clear idea of which users belong to which groups, it′s time to add a user to a validation list.

You only need to add a user to a validation list if they do not already have an AS/400 profile. You can use the AS/400 system′s normal user profile and password checking for users with an AS/400 profile. However, you may want to consider giving validation list userids and passwords even to those users who already have AS/400 user profiles. This ensures that any userid/password combination that is discovered by an untrustworthy person, or hacker, cannot be

used to compromise the integrity of the AS/400 operating system only to view documents stored on the Internet Connection Server for AS/400. Remember, though, that using SSL to encrypt session data renders the userid and password virtually indecipherable.

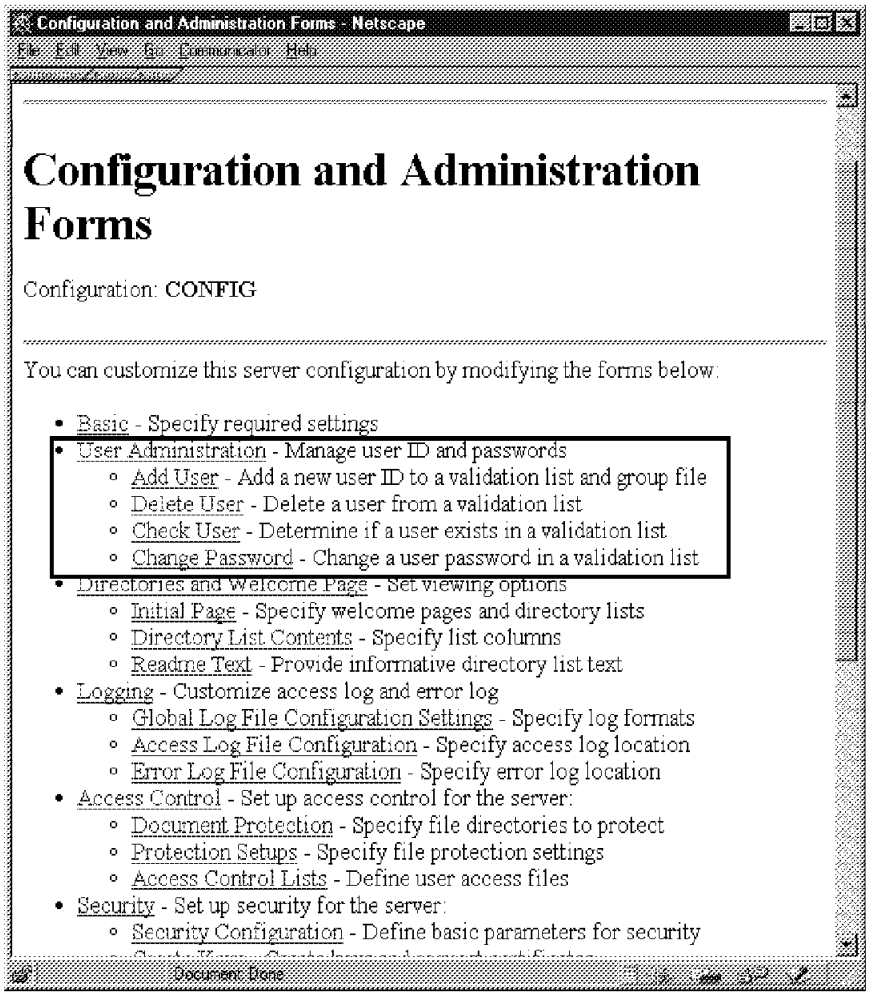


Figure 102. Configuration and Administration Forms Window

Click on **Add User** from the Configuration and Administration Forms window (see Figure 102).

We are going to add user Deidre to validation list VALLIST in library ITSOLIB and to the group techgrp, the details of which are stored in group file groupfl.grp in directory itsodir (see Figure 103 on page 129). The library and directory to which you add the validation list and groupfile respectively, must already exist. If the library and directory exist, the validation list and group file are added automatically.

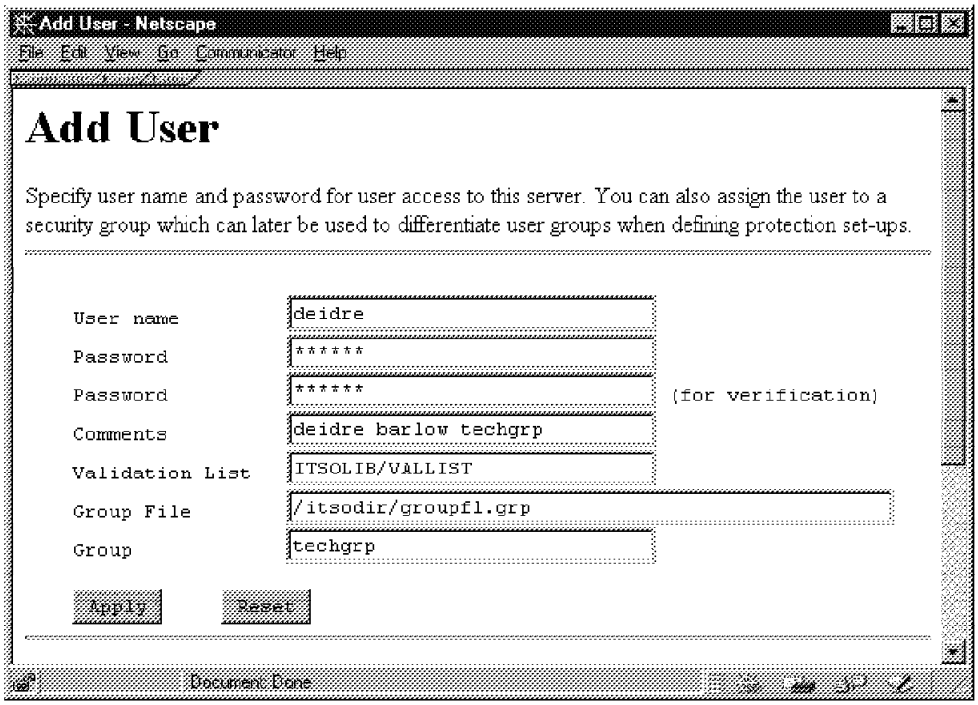


Figure 103. Adding User Deirdre

Note

It is important to remember that user names and passwords are case-sensitive when stored in the validation list. When a user enters their user name and password in the authorization pop-up, they are required to use the same case as used on this window. We recommend that you initially decide upon a common case, either uppercase or lowercase.

For detailed use of this form, and all other forms mentioned in this chapter, click on the **Help** icon as the help text for each form is quite comprehensive (see Figure 104 on page 130).

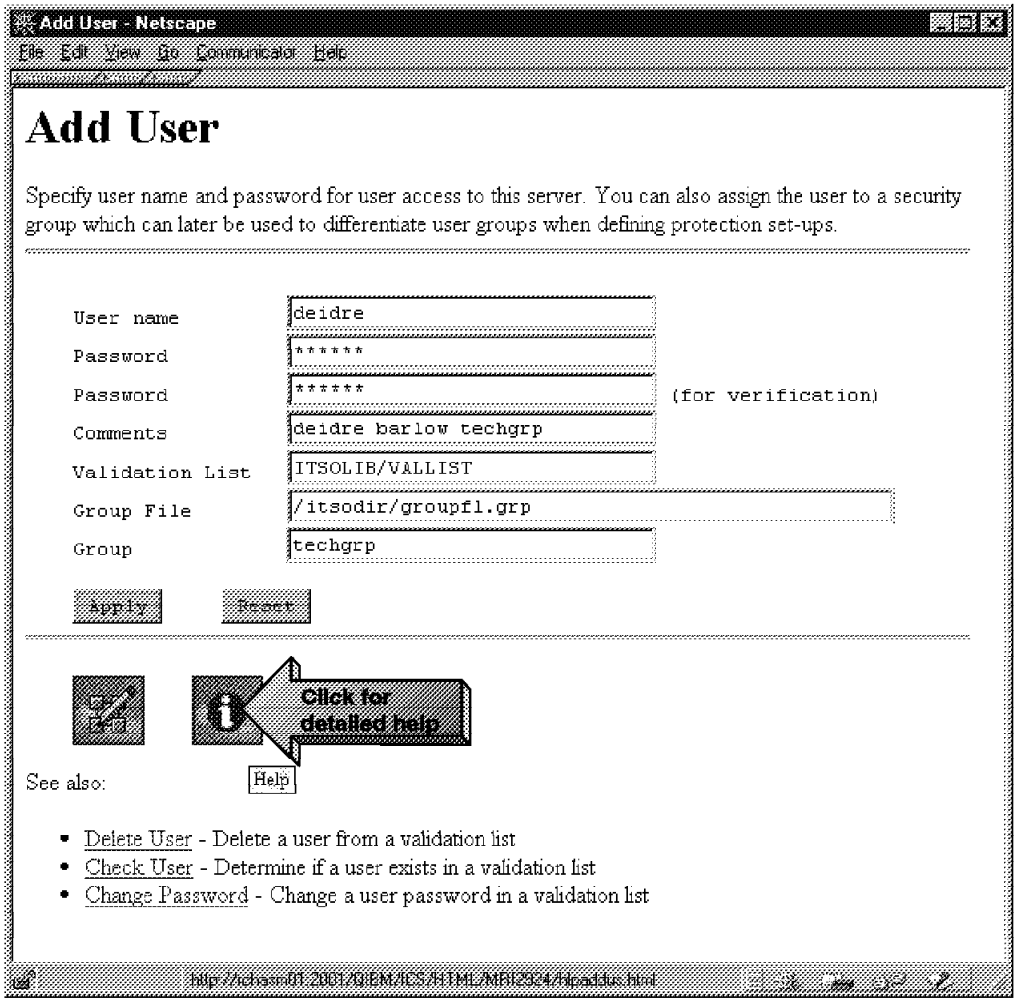


Figure 104. The HELP Icon

After filling in the fields correctly, click on **Apply**. If the operation is successful, you receive a confirmation window similar to the one shown in Figure 105 on page 131.

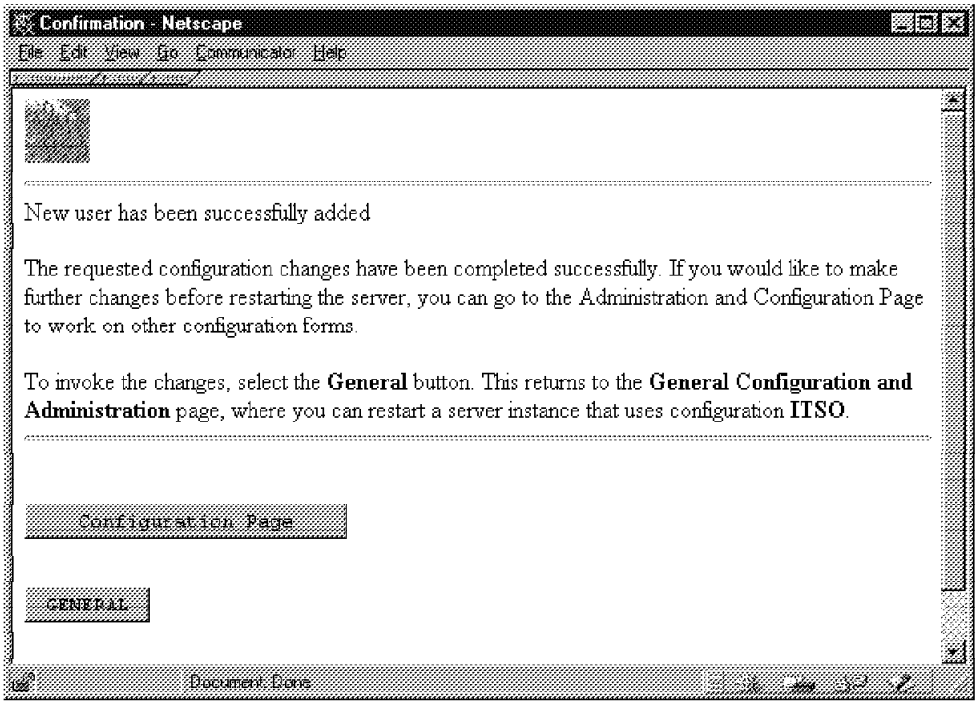


Figure 105. User Successfully Added

At this point, we have stored the user name and associated password in a validation list and also enrolled the user in a group. The group file is stored as a stream file in the root directory. It is possible, using Client Access or some other means of viewing text files on the AS/400 system to look at the contents of this file.

Figure 106 shows a group file, groupfl.grp, where user Deidre has been added to a group called techgrp.

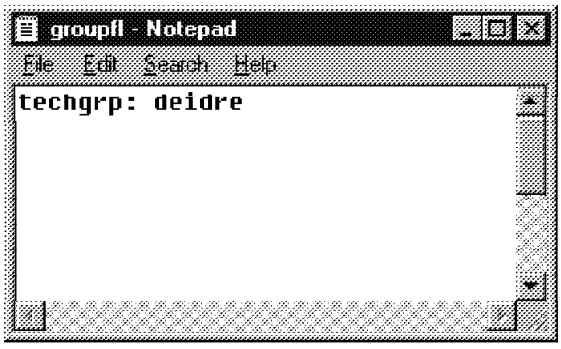


Figure 106. Group File Contents

There are many different ways to configure a group. It is possible to use user names, IP addresses, IP address ranges, and groups in different combinations.

Examples:

```
group1:  john, mary, bill@10.4.4.3
group2:  (david, steve)@10.5.*.*
group3:  All@www.ibm.com
group4:  group1, group2, group3
```

Let's examine the four examples more closely.

group1 Contains users john, mary, and bill. A request from bill is only accepted if bill makes the request from IP address 10.4.4.3. John and mary can make a request from any IP address.

group2 Contains users david and steve. A request from david or steve is only accepted if they make the request from an IP address beginning with 10.5.

group3 Contains all or any users in the domain www.ibm.com.

To use host names in the address template field, you must set the DNS-Lookup directive to **on**. Refer to Chapter 6, "Basic Internet Connection Server Configuration" on page 43 for details.

group4 Contains all those users that are contained within group1, group2, and group3.

You can create as many server group files as you need. Each one should be created in a separate text file. Within the group file, each line contains a group definition in the following format:

groupname:user1(,user2(,user3..)) where:

groupname

Any name you want to use to identify the group you are defining. This name can be used on subsequent group definitions within the same server group file.

user1(,user2(,user3..))

This can be any combination of user names, group names, or address templates. Separate each item with a comma *"*,*"*. Items can be grouped together using parentheses *"***( )***"*.

For user names to be valid, they must be defined in the validation list that the protection setup points to. Group names must be defined on previous group definition statements in the same group file.

There are three other User Administration options:

Delete User

Allows the administrator to delete a user name from a validation list and a group file.

Check User

Allows the administrator to check for the existence of a user name in a validation list. You must complete all fields before the user's existence is checked.

Change Password

Allows the administrator to change the password associated with a user in a validation list. The existing password is not required and the change takes place without the need to restart the server instance.

Having defined users, groups, and address templates using User Administration, we can start to implement access control over the Internet Connection Server for AS/400 resources.

8.4 Access Control

Internet Connection Server for AS/400 and Internet Connection Secure Server for AS/400 include enhanced access control support. These enhancements allow much greater control over who can access what on the server than is possible purely through resource mapping.

In our examples, we protect directories /itsoics/admin and /itsoics/tech in the ITSO company's directory structure. You may refer to the directory structure shown in Figure 107 as we go through the examples. It is desirable to have a well-laid out and documented directory structure. It is much easier to restrict access to, say, administration files if they are all contained in the same directory.

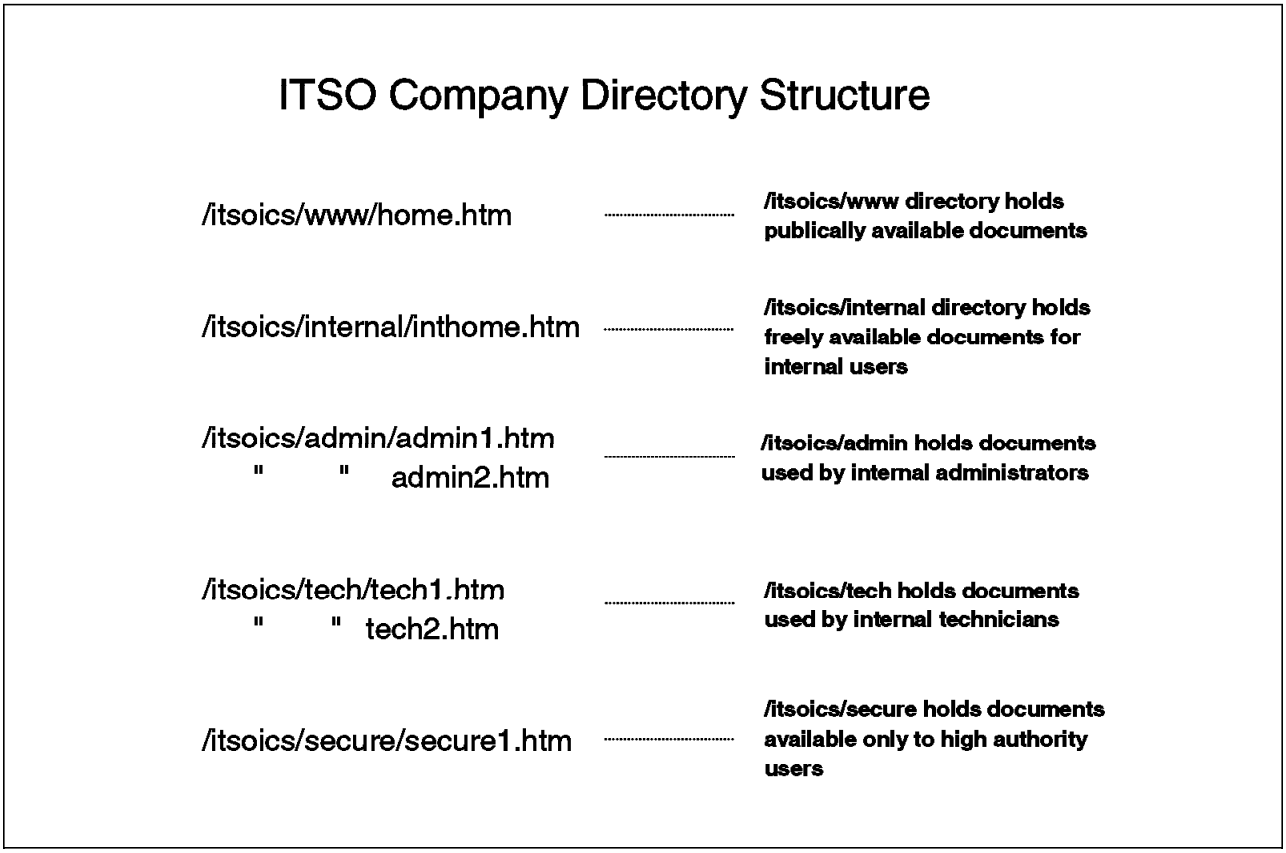


Figure 107. ITSO Company's Directory Structure

8.4.1 Protection Concepts

In this section, we discuss protection concepts at a high level before we go on to implement them. It is important that you understand these concepts before attempting to protect a document or directory.

Protection setups can be defined through the ADMIN server's browser interface or through the AS/400 WRKHTTPCFG command. In the following examples, we use the browser interface to define the protection setups. We then look at the configuration files changes using the WRKHTTPCFG command.

Note

Any server directives associated with a protection setup must appear in the HTTP configuration file before any Pass or Exec directive relating to the same incoming URL.

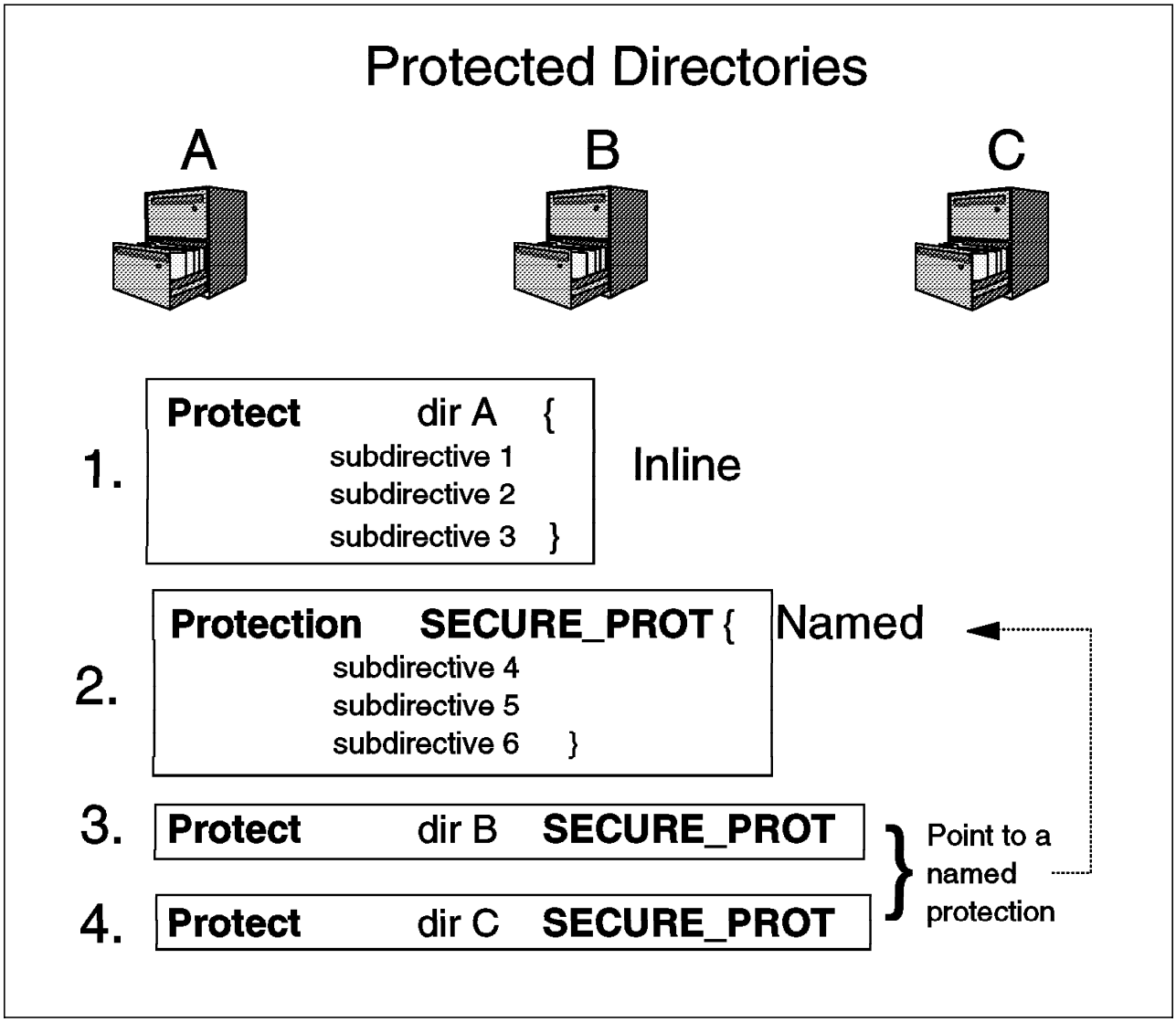


Figure 108. High-Level View of Protection

You can see in Figure 108 that we have three directories (A, B, and C) that we want to protect. We do this using Protect and Protection directives. We can also use the DefProt directive.

Protection subdirectives are the rules that tell the server how it should protect specified directories and files. There are two methods of defining these subdirectives:

Inline Where the protection subdirectives are defined within the Protect or DefProt directive. In this case, they apply to the URL template defined in the Protect or DefProt directive.

Named Where the Protect or DefProt directive points to a previously defined Protection directive. A Protection directive defines subdirectives in the same way as an inline protect directive but does not contain a URL template. It can be referenced, multiple times if necessary, by Protect and DefProt directives.

Note: We shall discuss the DefProt directive later in Section 8.4.11, “Default Protection (DefProt)” on page 148.

You can also see in Figure 108 on page 134 that we have four directives that activate protection in the following way:

1. This Protect directive is activated when a request is made to directory A. It is an inline Protect directive in that it contains subdirectives specific to the directory it is protecting.
2. This is a named Protection setup. It contains an identifying label, **SECURE_PROT**, and subdirectives. The subdirectives are not specific to any directory as, unlike a Protect or DefProt directive, it does not contain a request template. The request template and, therefore, the directory it is used to protect, is contained in the Protect directive or DefProt directive that references this named Protection.
3. This Protect directive is activated when a request is made to access directory B and points to a previously defined Protection setup using its label as a pointer. In this case, it points to the Protection setup **SECURE_PROT** and uses its subdirectives to protect directory B.
4. This Protect directive uses the Protection setup **SECURE_PROT** to protect directory C.

The actual format of an inline Protect or DefProt directive is as follows (see Figure 109).

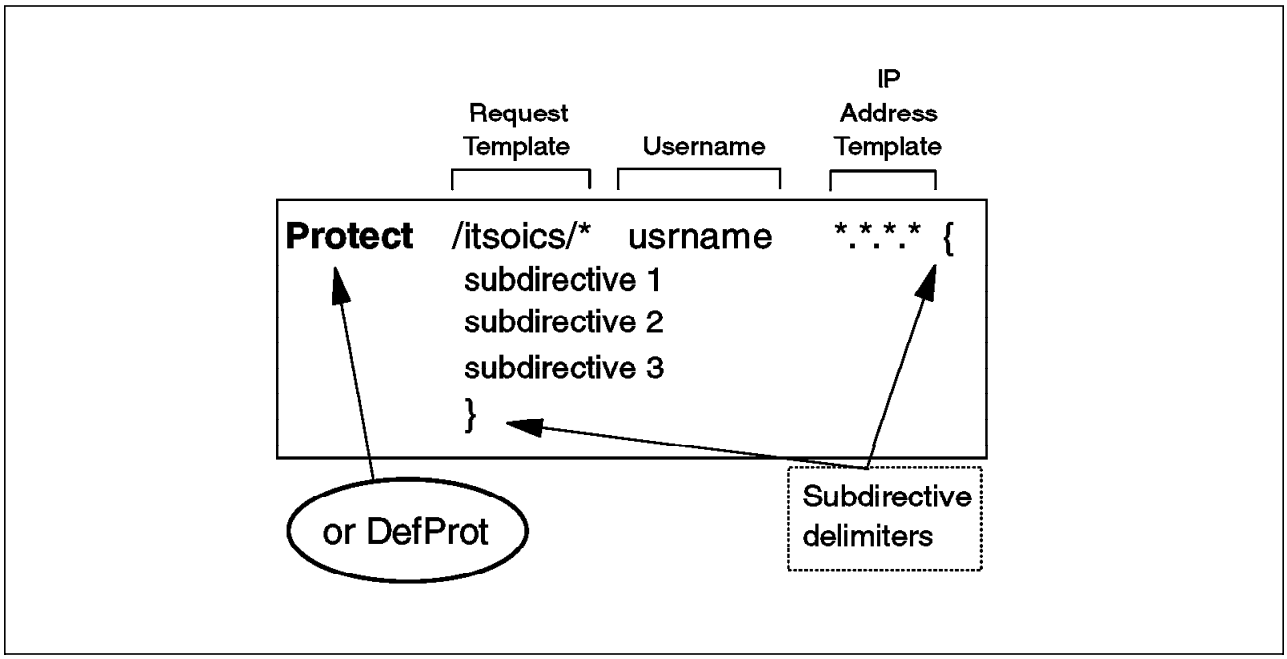


Figure 109. Inline Protect (or DefProt) Directive Format

Request Template

A template for requests that you want to associate with this directive. The server compares incoming client requests to the template and activates the directive if there is a match.

In the case of a DefProt directive, protection is not actually activated unless the request also matches a template on a subsequent Protect directive.

Username

This is an optional parameter. If used, it must be the name of a valid AS/400 user profile. It supplies the user profile to which the server should change when serving the request. It allows the use of AS/400 object access control. If not specified, the default AS/400 user profile for the server is used (QTMHHTTP).

IP Address Template

This is an optional parameter. If the server has multiple connections, you can use this parameter to specify an address template. The server uses the directive only for requests that come to the server on an address that matches the template. It is important to note that this template is compared to the IP address of the server, not the client. If no address is specified, the server can use this directive for all requests, regardless of the connection they come in on.

Subdirective Delimiters

Where a directive includes subdirectives, the last character of the line that contains the directive should be a left brace character (**{**). The subdirectives are placed on the following lines. To indicate the end of the subdirectives and the directive, there should be a right brace character (**}**) on a separate line following the last subdirective.

In Figure 110 on page 137, you can see a named DefProt directive. It does not contain subdirectives but it has an extra parameter, Label.

Label The Label refers to the name of a Protection setup that you want to associate with this directive.

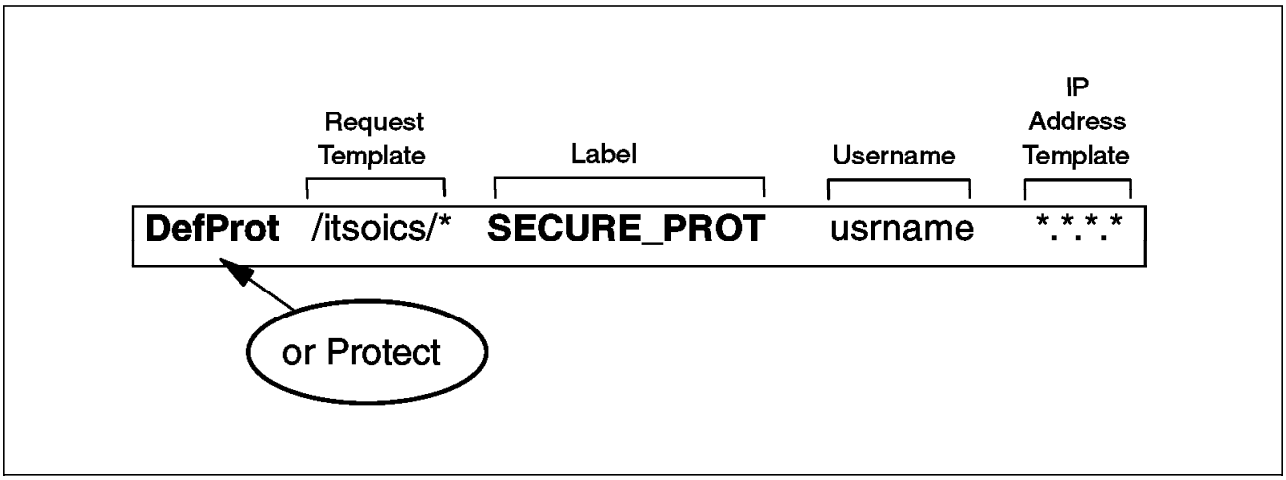


Figure 110. Named DefProt (or Protect) Directive Format

The last example shows the format of the Protection directive (see Figure 111).

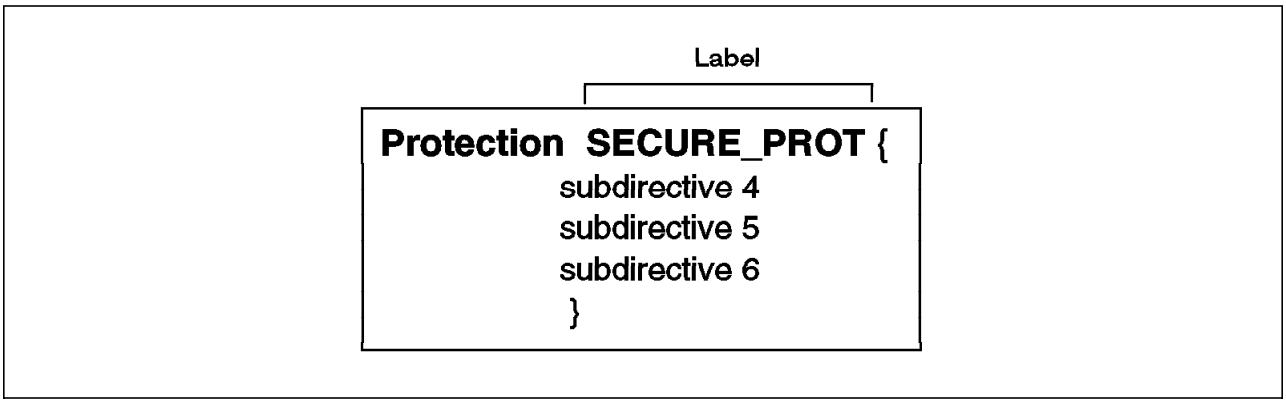


Figure 111. Protection Directive Format

You can see that the only parameter in a Protection directive is the Label parameter. The label, in this case, is the name to be associated with this Protection setup. The name can then be used by subsequent Protect or DefProt directives to point to this setup.

Now that we have an understanding of the relationship between Protect directives, DefProt directives, and Protection setups, let's define some directives using the ADMIN Server's browser interface.

8.4.2 Adding an Inline Protect Directive

When you add an inline protect directive, you specify the directory you want to protect and then the subdirectives that define how the directory is to be protected. Specifying the directory you want to protect is done through the Document Protection form. Once completed, you are presented with the Protection Setups Form where you can add the protection subdirectives. First, though, we go to the Document Protection Form.

8.4.3 Document Protection - Inline

From the Configuration and Administration Forms window, click on **Document Protection**.

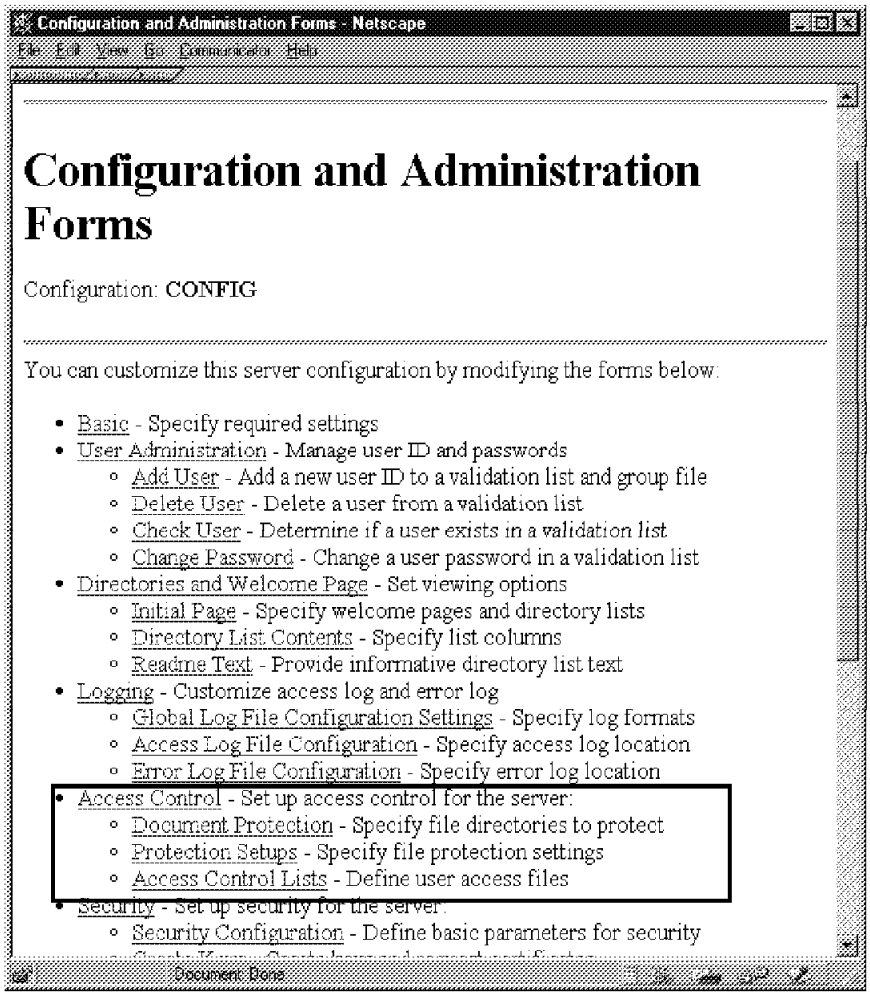


Figure 112. Configuration and Administration Forms Window

We are going to add an inline Protect directive to protect directory /itsoics/admin.

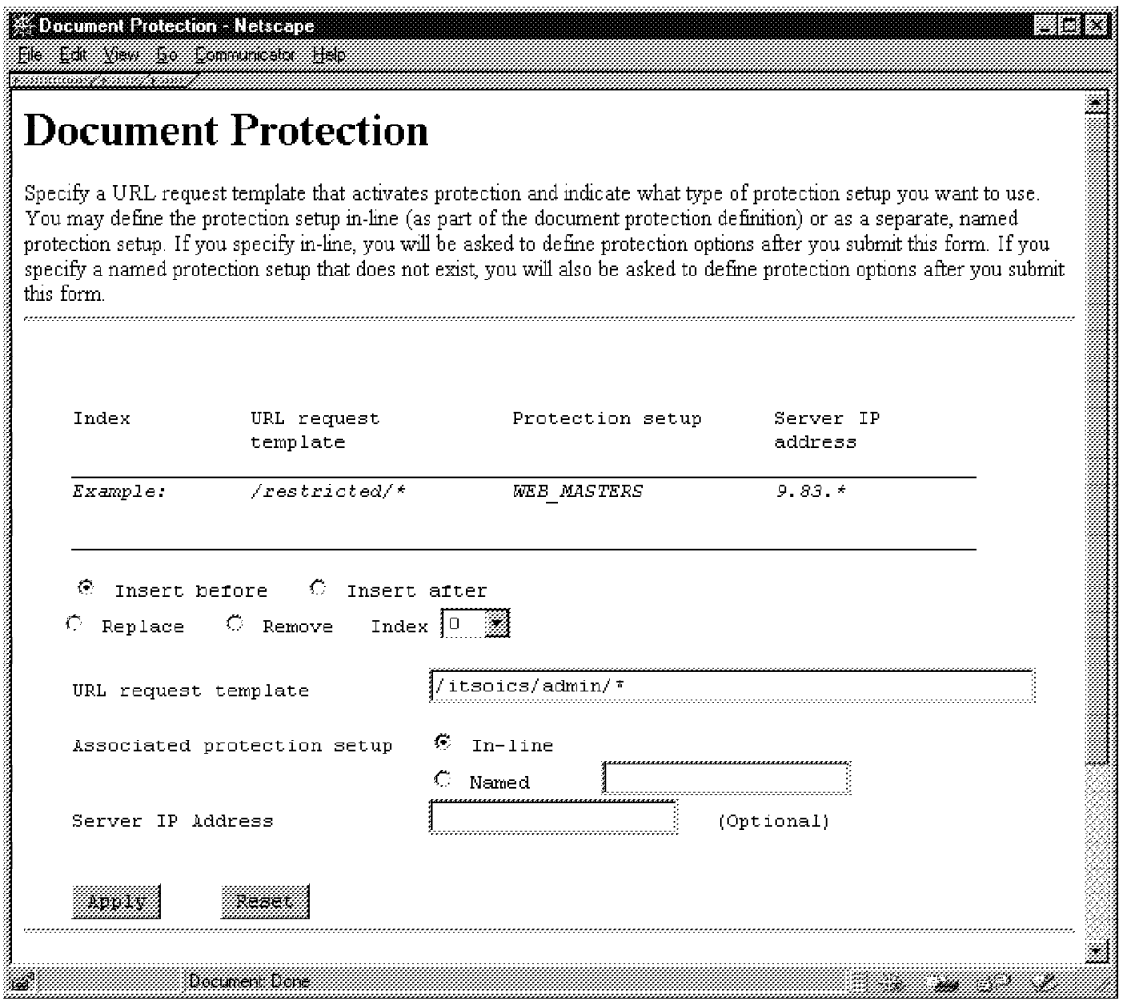


Figure 113. Specifying URL Request Template for Protection

You can see in Figure 113 that we have entered /itsoics/admin/* as the **URL request template**. Any requests to directory /itsoics/admin or subdirectories within this directory match this template and so activate this protect directive. For the **Associated protection setup**, we have selected the In-line box. This results in a window being presented where we define the inline subdirectives. We chose not to complete the optional parameter **Server IP Address**. If we want this protection setup to only apply to requests received on a specific server interface, we specify the IP address of that interface here. This can be a useful feature if the server has several interfaces, each with a unique IP address, and you want the server to behave differently on each IP address. For a detailed explanation of this form, click on the **Help** icon.

The **Insert before** button is selected by default. Using the **Index** pull-down, it is possible to add a directive either before or after a specified index number. In this example, it is irrelevant because it is the first directive. However, because the order of the HTTP configuration file is important, it may be necessary at a later date to place directives carefully.

When the form has been completed, click on **Apply** to receive the Protection setup form (See Section 8.4.4, “Protection Setup - Inline” on page 140) where we configure the inline subdirectives.

8.4.4 Protection Setup - Inline

Having specified that we want to protect directory /itsoics/admin, we need to tell the server how to protect it. For this, we need to fill out the Protection Setup form.

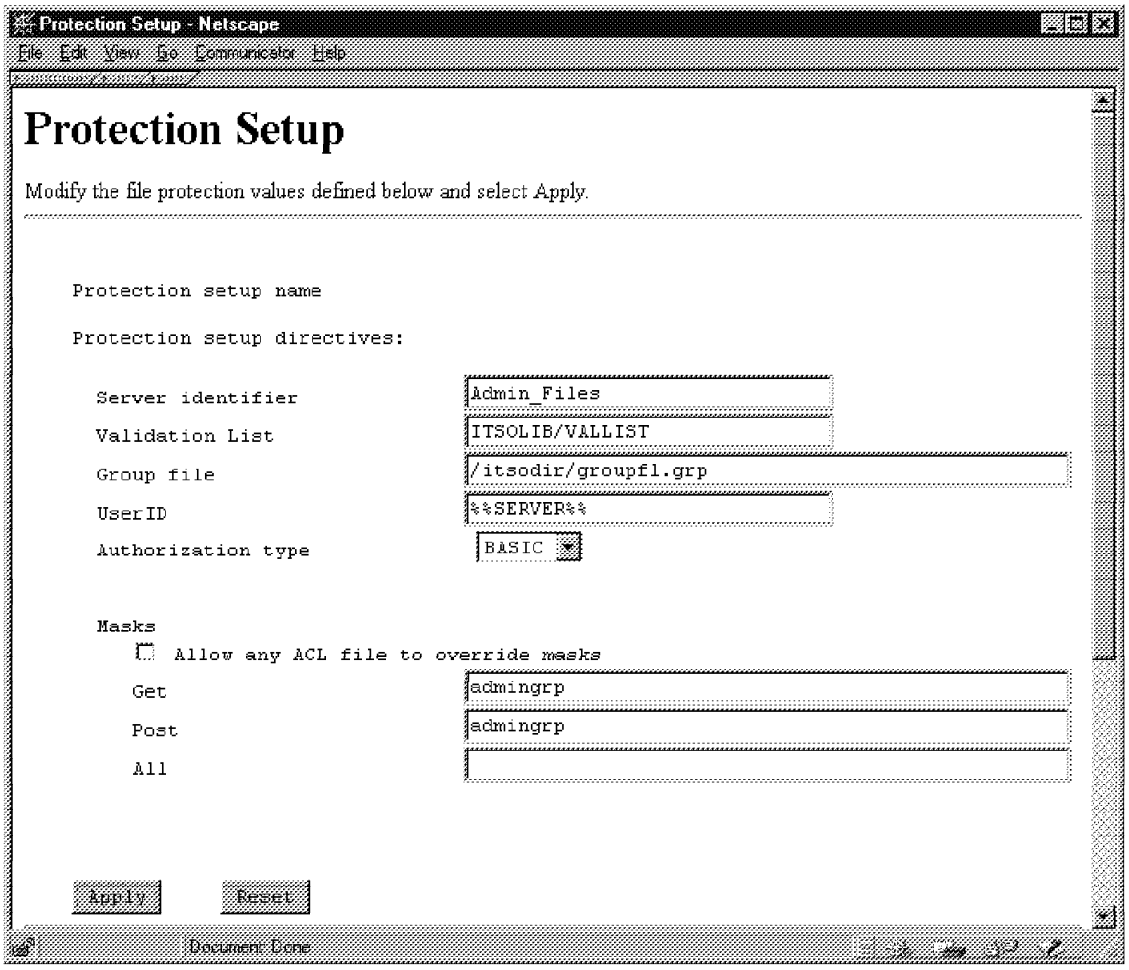


Figure 114. Specifying Protection Values

You can see in Figure 114 that we specified Admin_Files for the **Server Identifier**. This is shown on the userid/password pop-up and can be used as a reminder to people when they enter their userid and password (see Figure 115 on page 141). This is its only use and does not need to be a real system name.

Figure 115. Userid and Password Pop-up

We also specified:

- **Validation list** ITSOLIB/VALLIST: This is where our userids and passwords were stored during the Add User phase of User Administration (see Section 8.3.1, “Adding a User” on page 127).
- **Group file** /itsodir/groupfl.grp: This is where our groups were stored during the Add User phase of User Administration (see Section 8.3.1, “Adding a User” on page 127). It is used by access control to resolve the userids contained within a group.
- **UserID** %%SERVER%%: This is the AS/400 user profile that the server will switch to while completing the request. %%SERVER%% is the profile of the server, which is QTMHHTTP by default.
- **Authorization type** BASIC: This value is used if any part of the protection setup uses user names and password protection. It is the default value.
- **Masks:**
  - **Allow any ACL file to override masks**. We left this box unchecked as we do not want our protection rules taken from an ACL file. ACL files are discussed in Section 8.4.12, “Access Control Lists” on page 151.
  - **Get** admingrp: This allows any user name in the group called admingrp to perform HTTP GET requests on the protected resource.
  - **Post** admingrp: This allows any user name in the group called admingrp to perform HTTP POST requests on the protected resource.
  - **All**: We left this blank. This field allows the authorization of requests not covered in the GET or POST fields.

When we were satisfied with the values in the form, we clicked **Apply** to add the directive to the HTTP configuration file. Once the server instance has been restarted from the General Configuration and Administration form, protection is active on directory /itsoics/admin.

8.4.5 Using AS/400 User Profiles

As mentioned previously, it is possible to use AS/400 user profiles and passwords for access control. To implement this, we specify %%SYSTEM%% on the Protection Setup form instead of specifying a validation list. This tells the server that userids and passwords entered should be resolved using the system password file instead of a validation list.

Note: Although it is possible to use AS/400 user profiles and passwords, it is not possible to enter them into a group. The user names within a group file must point to a validation list.

Figure 116 shows adding a protection setup that uses AS/400 user profiles.

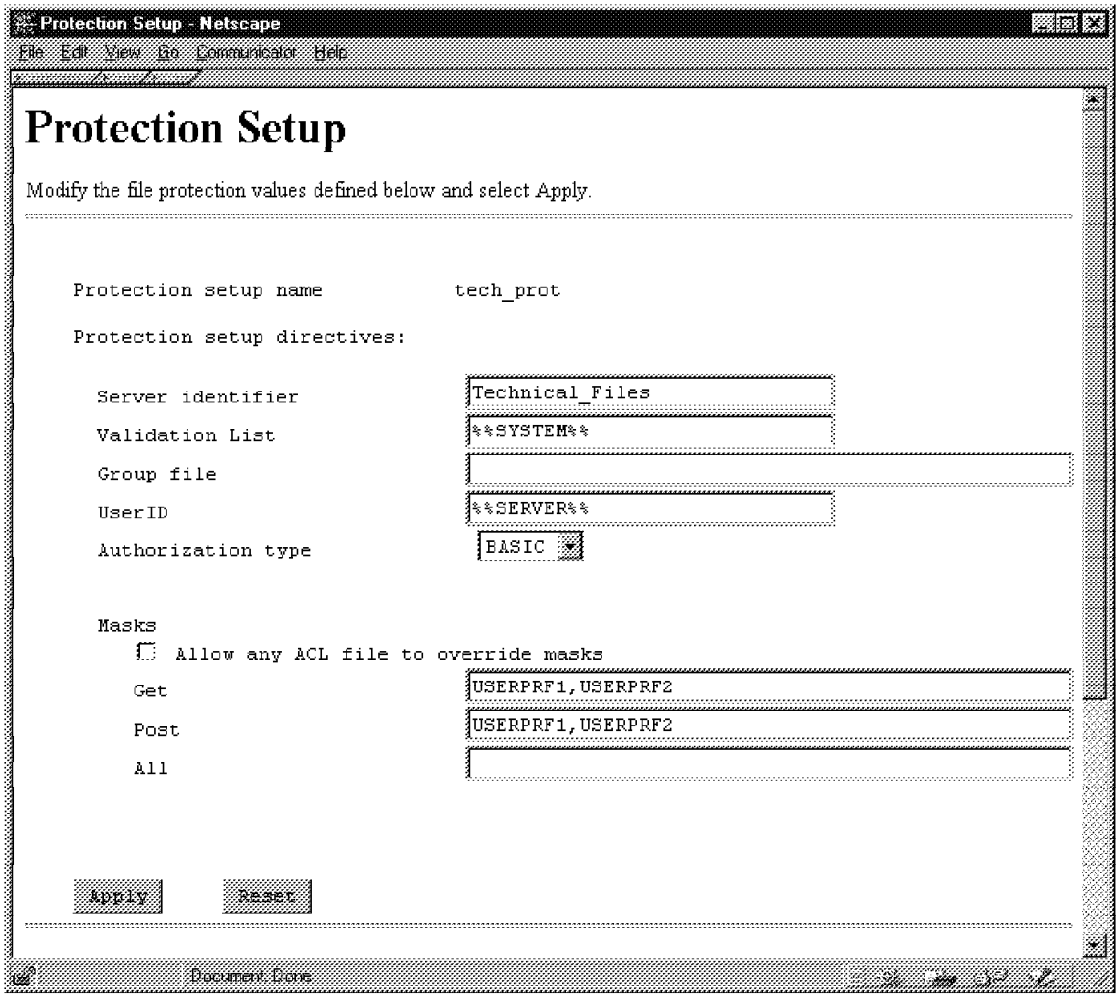


Figure 116. Protection Setup Using AS/400 User Profiles

You can see in Figure 116 that we have specified a validation list of %%SYSTEM%% so that user ids are checked against valid AS/400 user profiles and, for the masks, we specified that USERPRF1 and USERPRF2 can use methods GET and POST so long as they are valid user profiles and the associated password is entered.

8.4.6 HTTP Configuration - Inline

We can use the WRKHTTPCFG command to display the changes to the HTTP configuration.

```
                HTTP Configuration File
0030     Protect  /itsoics/admin/*  {
0040           GroupFile    /itsodir/groupfl.grp
0050           PasswdFile   ITSOLIB/VALLIST
0060           ACLOverride  Off
0070           PostMask     admingrp
0080           GetMask      admingrp
0090           AuthType     BASIC
0100           ServerID     Admin_Files
0110           UserID       %%SERVER%%
0120     }
```

Figure 117. Inline Protect Directive Added to HTTP Configuration File

Figure 117 shows the results of implementing the inline protect directive for directory /itsoics/admin using the forms shown in Figure 113 on page 139 and Figure 114 on page 140.

8.4.7 Adding a Named Protection

In the previous section, we added protection using inline protection. In this section, we look at adding protection using named protection. Having created the named protection, we use it to protect a directory. Named Protection setups are useful if you need to protect several directories in the same way.

8.4.8 Protection Setup - Named

To add a Named Protection Setup, go to the Configuration and Administration Forms window and click on **Protection Setups**. The Protection Setup List form is shown (see Figure 118).

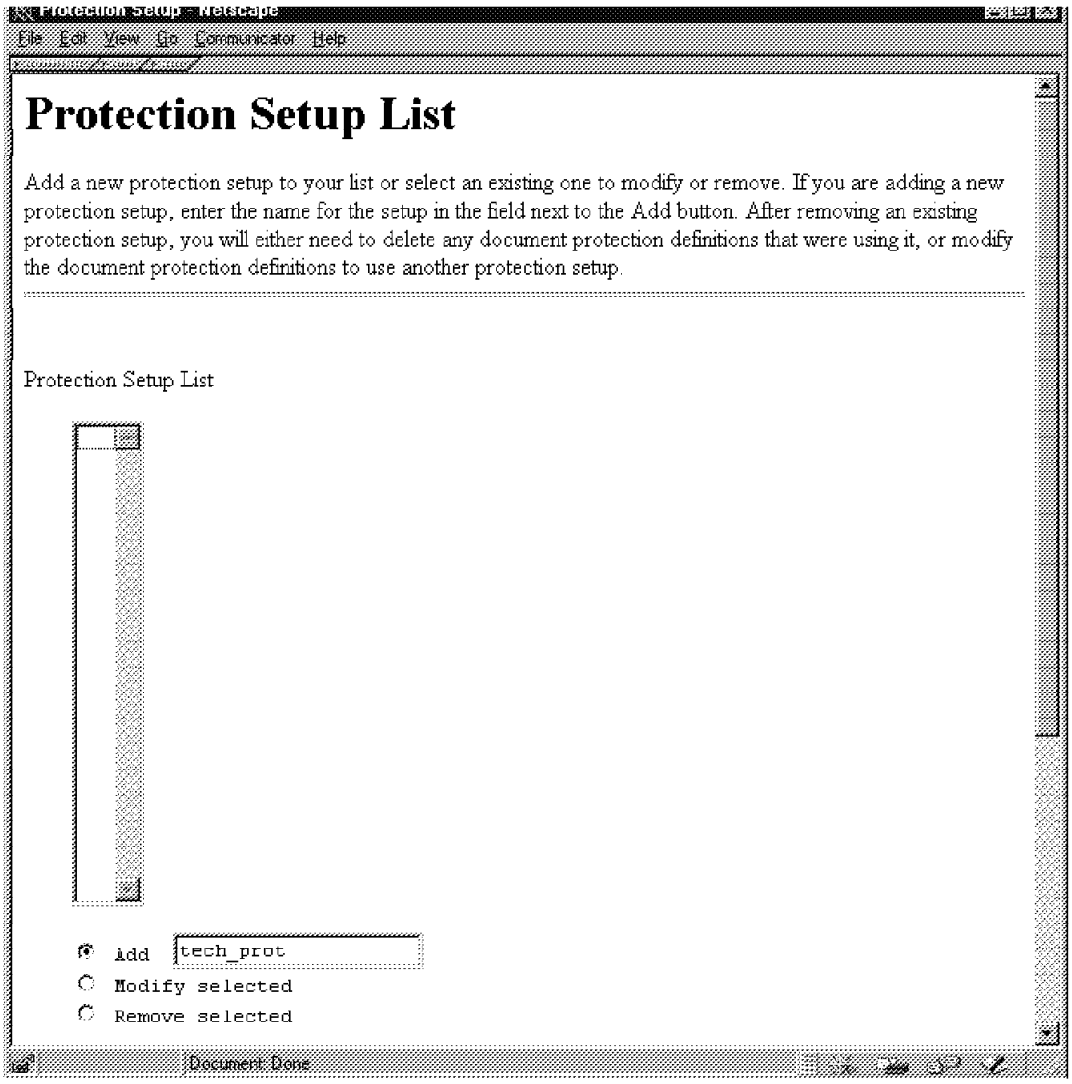


Figure 118. Protection Setup List Form

From this form we can add, modify, or remove named protection setups. We are adding a named protection called tech_prot. Having added a name by which the protection setup is referenced, select the **Add** box and click on **Apply**. The Protection Setup form is shown. This time, however, you can see that the protection setup has a name, tech_prot. Here we specify the subdirectives that the protection setup is to use (see Figure 119 on page 145).

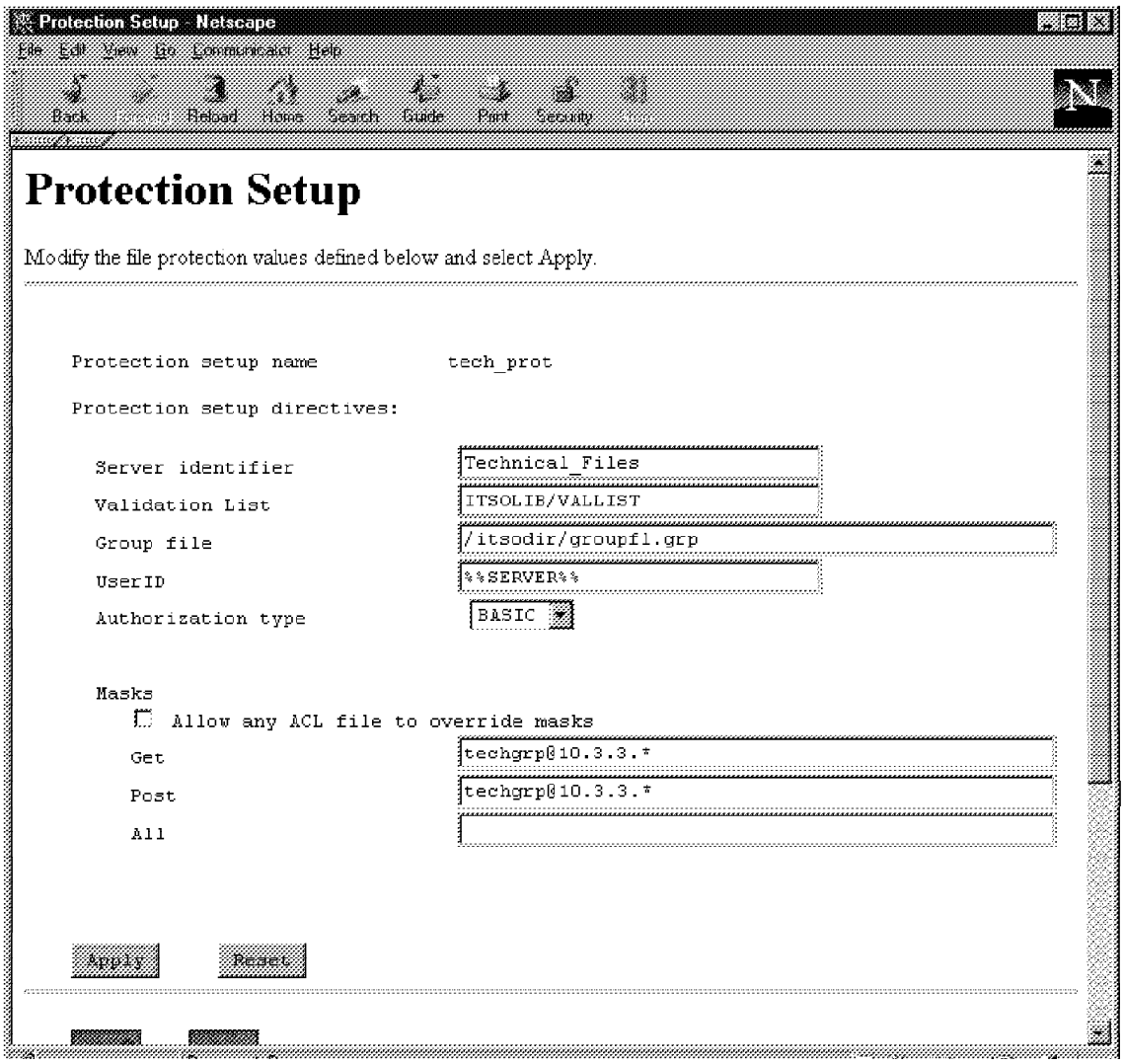


Figure 119. Adding Subdirectives to a Named Protection Setup

We specified Technical_Files as the Server Identifier to act as a reminder on the userid/password pop-up.

We also specified:

- **Validation list** ITSOLIB/VALLIST: This is where our userids and passwords were stored during the Add User phase of User Administration (see Section 8.3.1, “Adding a User” on page 127).
- **Group file** /itsodir/groupfl.grp: This is where our groups were stored during the Add User phase of User Administration (see Section 8.3.1, “Adding a User” on page 127). It is used by access control to resolve the userids contained within a group.
- **Masks**: We added techgrp@10.3.3.* to both GET and POST masks. This allows anyone from the group techgrp to access the protected directory providing they enter a valid userid/password and they are making the request from an IP address of 10.3.3.* (where * is any value).

Once the form is completed, click on **Apply** to add the named protection to the HTTP configuration file.

As we explained previously, a named protection is of no use unless it is being referenced by a Protect or DefProt directive. In the following section we shall use the Document Protection form to specify a directory to protect and reference the named protection, tech_prot.

8.4.9 Document Protection - Named

To add Document Protection using a named protection setup, click on **Document Protection** from the Configuration and Administration Forms window. The Document Protection form is shown again as in Section 8.4.2, “Adding an Inline Protect Directive” on page 138. This time, however, we are going to specify directory /itsoics/tech as the directory to protect and we select named protection, not inline protection (see Figure 120).

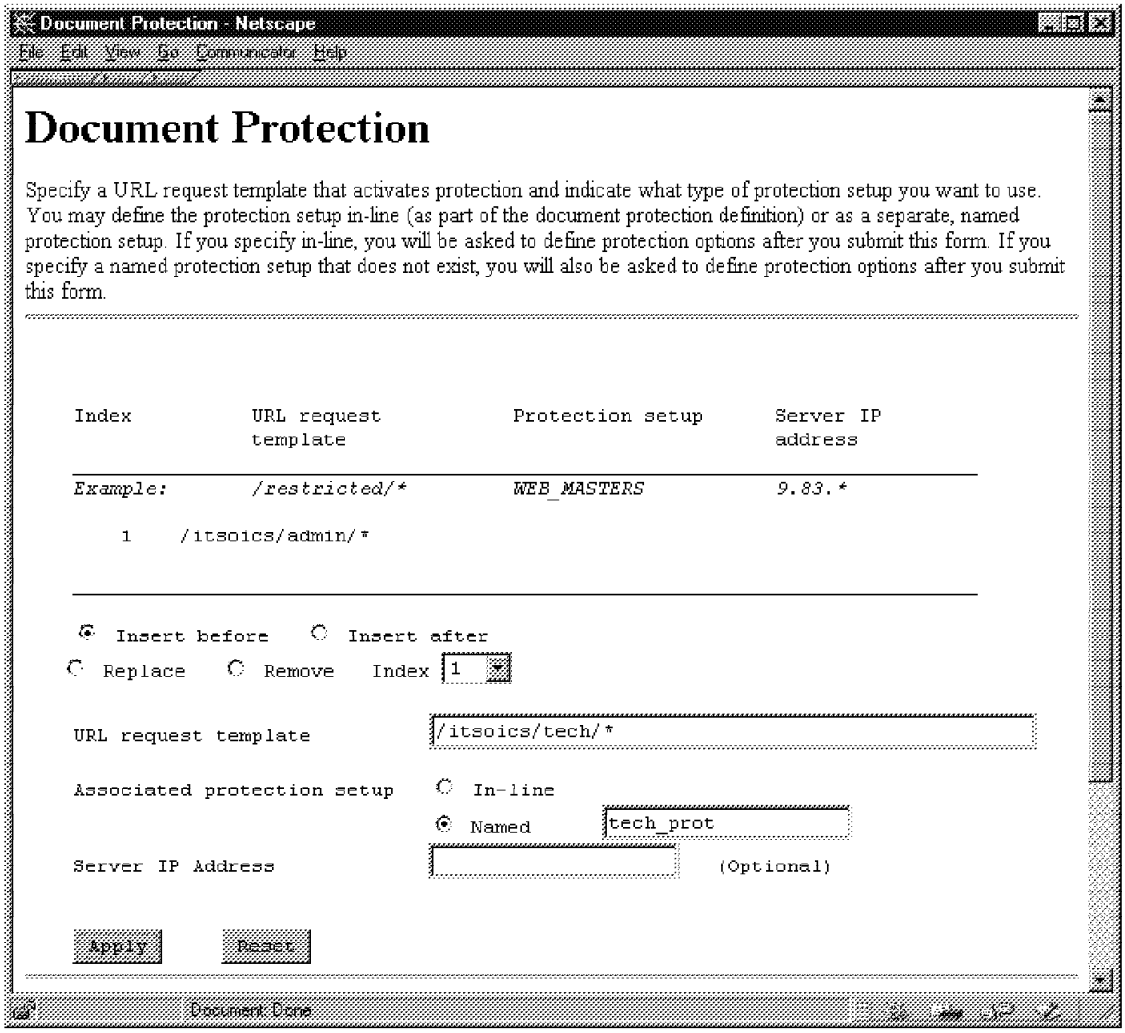


Figure 120. Protecting a Directory with a Named Protection

We have entered /itsoics/tech/* as the URL request template. This protection setup is activated when any request is made to directory /itsoics/tech or any of its subdirectories. To reference our previously defined named protection, we selected **Named** and entered the name of the protection setup tech_prot. When

you have completed the form, click on **Apply** and you receive the Protection Setup form, which describes the named protection you referenced. This allows you to make any changes to the named protection setup, in our case it was tech_prot (see Figure 119 on page 145). If you are satisfied with all the values, click **Apply** to add the Protect directive.

Once the relevant server instance has been restarted from the General Configuration and Administration form, protection is active on directory /itsoics/tech.

8.4.10 HTTP Configuration - Named

We can use the WRKHTTPCFG command to display the changes to the HTTP configuration.

```
           HTTP Configuration File
0030     Protection  tech_prot  {
0040           GroupFile    /itsodir/groupfl.grp
0050           PasswdFile   ITSOLIB/VALLIST
0060           ACLOverride  Off
0070           PostMask     techgrp@10.3.3.*
0080           GetMask      techgrp@10.3.3.*
0090           AuthType     BASIC
0100           ServerID     Technical_Files
0110           UserID       %%SERVER%%
0120     }
"""
"""
"""
"""
0220     Protect  /itsoics/tech/*  tech_prot
```

Figure 121. Protect Directives in HTTP Configuration File

Figure 121 shows a part of the resultant HTTP configuration file that was created when we added a named protection setup, tech_prot, and specified a directory to protect, /itsoics/tech, using the forms in Figure 119 on page 145 and Figure 120 on page 146.

8.4.11 Default Protection (DefProt)

Until now, we have talked about Protect and Protection directives. There is another directive, Default Protection or DefProt.

The Defprot directive is used to associate a default protection setup with a request that matches a template. This can be useful, for example, when we want to define default protection for a directory that is overridden for some subdirectories within the directory with a higher level of protection. The main difference between a DefProt and a Protect directive is that when an incoming URL matches the template on a DefProt directive, it is not activated unless it also matches the template on a subsequent Protect directive. Consider the example shown in Figure 122.

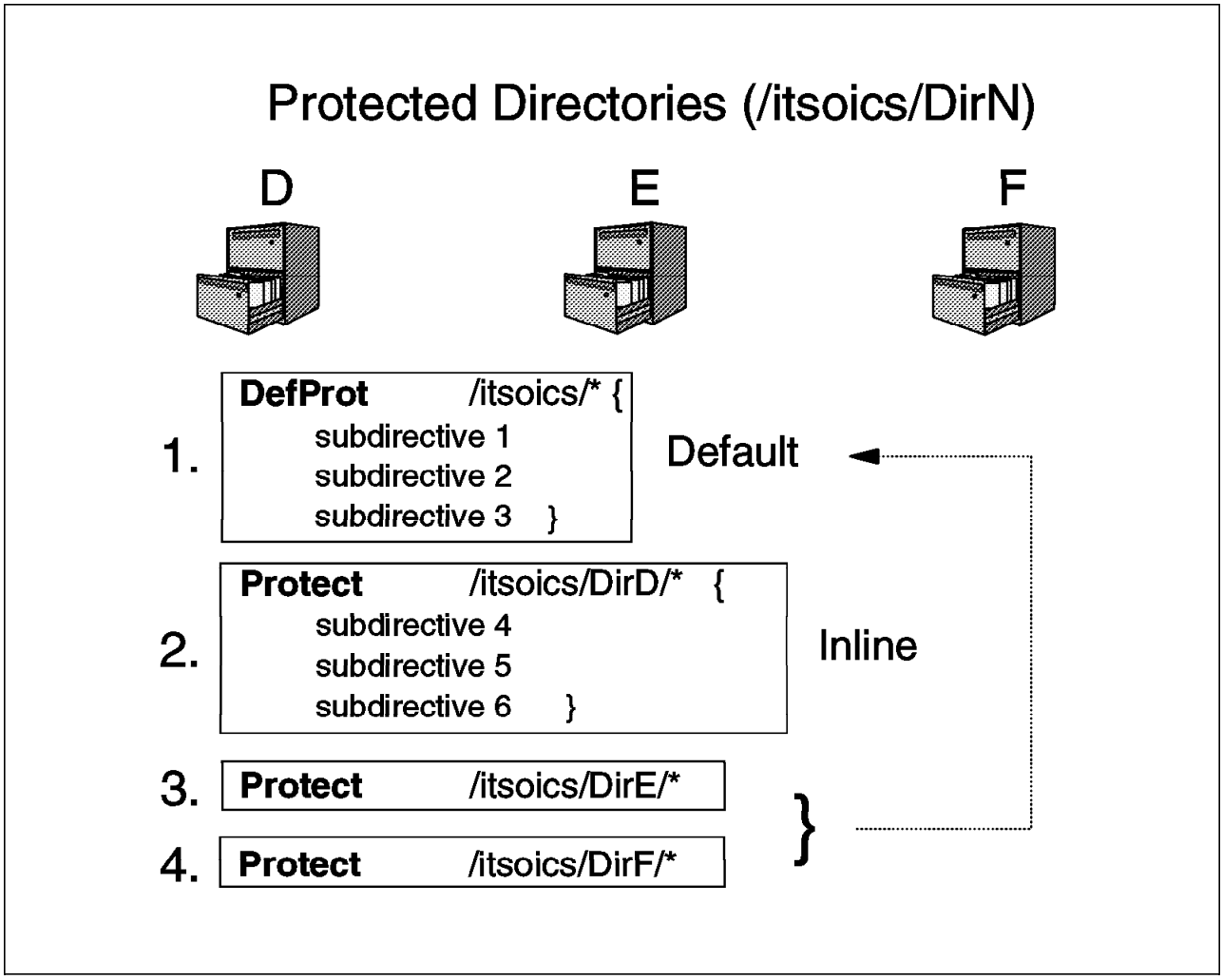


Figure 122. Default Protection Example

You can see that we have configured four directives. We describe their meaning in the following list:

1. This is the DefProt Directive. The request template /itsoics/* means that this directive is activated if any incoming URL matches this template *and* the template of any subsequent Protect directive that does not contain, or does not point to, any subdirectives explicitly.

2. This is an inline Protect directive that we have explained previously in Section 8.4.2, “Adding an Inline Protect Directive” on page 138. It protects directory /itsoics/DirD with inline subdirectives.
3. This Protect directive protects directory /itsoics/DirE but contains no subdirectives or pointer to subdirectives. However, if the server receives an incoming URL that matches this template as well as the template on the previous DefProt directive, protection is activated.
4. This Protect directive also activates the DefProt directive if we receive the incoming URL /itsoics/DirF.

Remember, the DefProt directive is ignored unless there is an ″associated″ Protect directive with no subdirectives and an incoming URL matches the URL templates on both of them.

8.4.11.1 Adding a DefProt Directive

Adding a DefProt directive is unlike adding a Protect directive or a Protection Setup in that there is currently no way of doing so purely through the ADMIN Server′s browser interface.

The following steps describe a method of adding an inline DefProt directive:

1. Follow the procedure for adding an inline Protect directive as we discussed in Section 8.4.2, “Adding an Inline Protect Directive” on page 138.

   When you have completed both forms, Document Protection and Protection Setup, you have added a new Protect Directive to the HTTP configuration file. In the next step, we change this (using native AS/400 displays and CL commands) to a DefProt directive.

2. Go to a native AS/400 command line and enter the CL command:

   ```
   WRKHTTPCFG config_name
   ```

   The HTTP configuration is displayed (see Figure 123).

```
                   Work with HTTP Configuration
                                                      System:  ITSO
  Configuration name . . . . . . . :   ITSO

Type options, press Enter.
  1=Add   2=Change   3=Copy   4=Remove   5=Display   13=Insert

      Sequence
Opt    Number    Entry

2_     00120     Protect  /itsoics/admin/*  {
__     00130           GroupFile    /itsodir/groupfl.grp
__     00140           PasswdFile   ITSOLIB/VALLIST
__     00150           ACLOverride  On
__     00160           PostMask     admingrp
__     00170           GetMask      admingrp
__     00180           AuthType     BASIC
__     00190           ServerID     Admin_Files
__     00200           UserID       %%SERVER%%
__     00210     }
```

Figure 123. Work with HTTP Configuration Display

3. Page down until you find the Protect directive added.

4. Type a "**2**" on the related line and press Enter.

   The Change HTTP Configuration Entry display is shown.

5. Overtype the text "**Protect**" with the text "**DefProt**" and press Enter to return to the Work with HTTP Configuration display.

```
               Change HTTP Configuration Entry
                                                    System:   ITSO
Sequence Number  . . . . . . . . . . . :   00120
Entry  . . . . . . . . . . . . . . . . .   DefProt  /itsoics/*  {
```

Figure 124. Change HTTP Configuration Display

6. Add associated Protect directives by typing a "**1**" (Add) and a sequence number; then press Enter. The Add HTTP Configuration Entry display is shown. Type the Protect directive on the Entry line (as shown in the example in Figure 125) and press Enter. Repeat this process for all the Protect directives you want to use the DefProt directive. When completed, the server instance should be restarted to implement the new directives.

```
               Add HTTP Configuration Entry
                                                    System:   ITSO
Sequence Number  . . . . . . . . . . .   00201
Entry  . . . . . . . . . . . . . . . .   Protect /itsoics/secure/*
```

Figure 125. Add HTTP Configuration Entry

   It is important that the sequence number of this Protect directive is greater than the sequence number of the added DefProt directive.

Note: It is essential that you change the Protect directive to a DefProt directive; otherwise, the Protect directive is activated immediately if a matching URL is found causing unpredictable results, whereas the DefProt directive is ignored until a subsequent Protect directive is found with a matching URL template. This is another example of the importance of getting the order of the HTTP configuration file correct.

It is possible to add subdirectives to a DefProt directive either inline or through a named protection (see Figure 126).

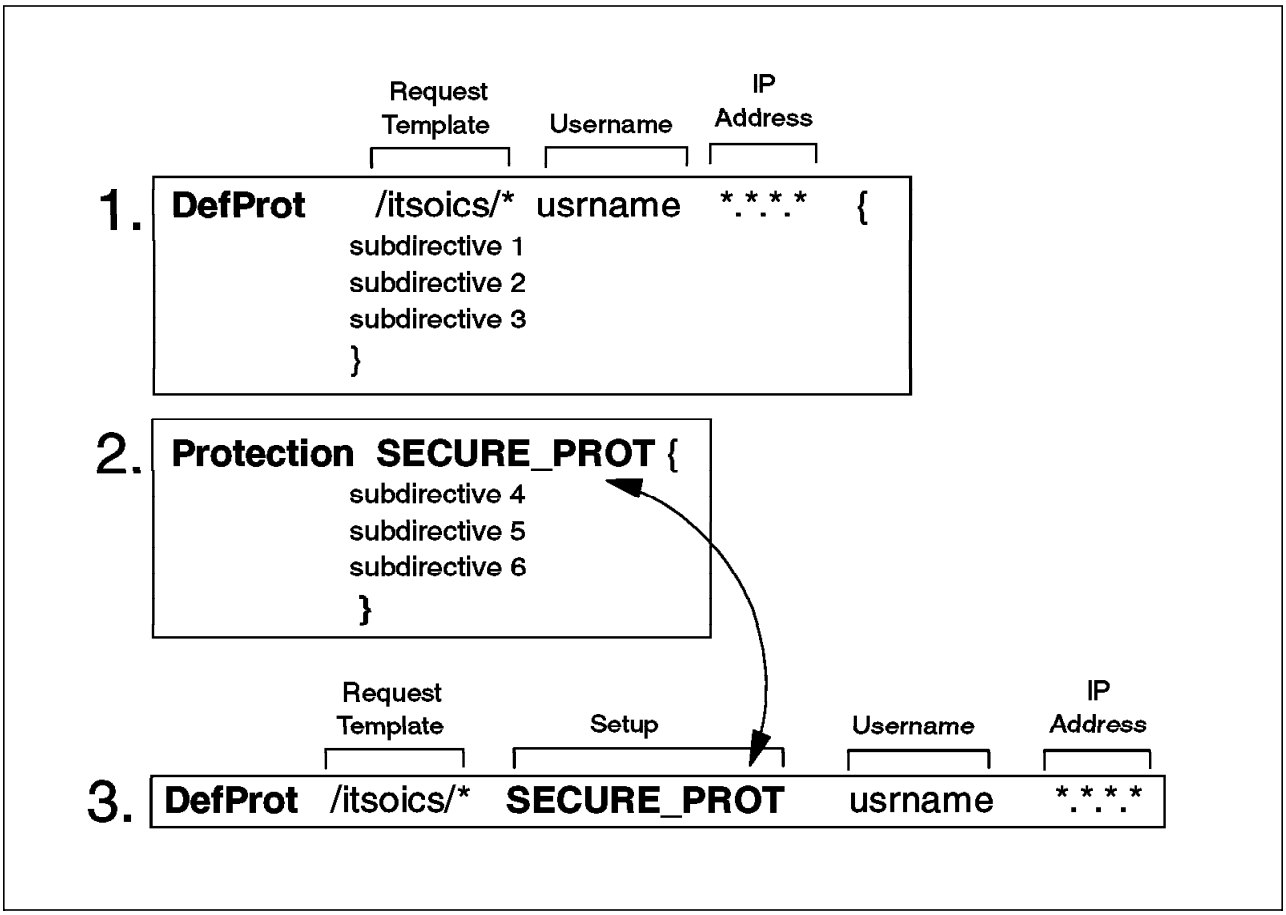


Figure 126. DefProt Inline and Named

You can see in Figure 126 that, as with a Protect directive, a DefProt directive can be either inline as in Example 1 or named as in Example 3, where Example 3 references the existing Protection in Example 2.

The DefProt subdirective can be used as an efficient way of adding a default protection to many directories.

8.4.12 Access Control Lists

Until now, we have used access control to protect the system down to the directory level. An access control list provides a method of protecting individual files (or types of files) within a directory. The access control list (ACL) is defined in a file. This file must be named **www_acl** and be placed in the directory to be protected.

Normally, the mask subdirectives in the protection setup define the first level of access control and then the ACL file further limits access. However, if we want all control to come from the ACL file, we can use the ACLOverride subdirective with a value of ON in the protection setup. The ACLOverride subdirective can be turned on or off from the Protection Setup form by checking the **Allow any ACL file to override masks** box (see Figure 114 on page 140).

We can use ACL files in the following integrated file systems:

- Root file system
- QLanSvr
- QOpenSys
- QDLS

Let's take a high-level look at an access control list.

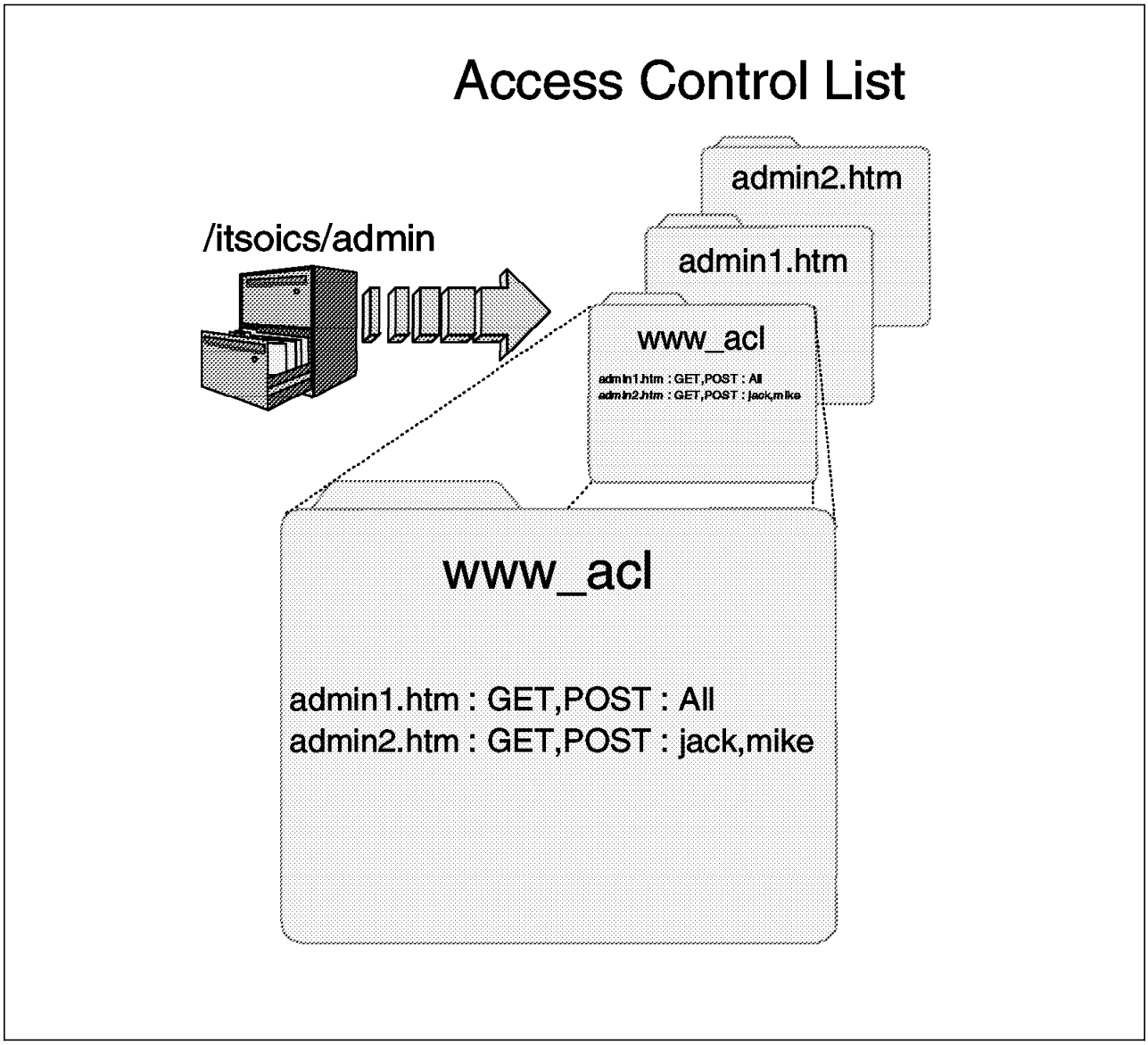


Figure 127. Access Control List - Overview

You can see in Figure 127 that we have added an ACL file to the /itsoics/admin directory. This is so that we can restrict access to the individual files, admin1.htm and admin2.htm, within the directory.

Remember that in Section 8.4.2, "Adding an Inline Protect Directive" on page 138, we added an inline protect directive that allowed anyone in the group admingrp to access any file contained in the /itsoics/admin directory. In this section, we further restrict access to the files within this directory.

Within the ACL file, each line contains a rule limiting access based on:

- File name
- HTTP method
- Authorized users, groups, or addresses.

The effects of using an access control list varies depending on whether the **Allow any ACL file to override masks** box is selected on the protection setup form or not (see Figure 114 on page 140):

1. **Allow any ACL file to override masks** unchecked:

   In this case, the masks defined in the protect directive implement a particular level of protection (in our case, only users from the group admingrp can access the directory /itsoics/admin). The ACL then further restricts access based on the rules contained within it. In the example in Figure 127 on page 152, we say that the documents in the directory are protected in the following way:

   - All users can GET or POST to admin1.htm so long as they are in the group called admingrp. This is because the GET and POST masks in the original protect directive restricted access to the group admingrp and the ACL file rules state that **All** users can GET and POST to that file. The ACL file imposed no additional restrictions.
   - Only users Jack and Mike can GET or POST to admin2.htm so long as they are in the group called admingrp. User Mike is not in the group admingrp and so he is not authorized to the document. User Jack is in the group admingrp and can, therefore, access the document. The ACL file imposed restrictions in addition to those specified in the protect directive′s masks.

2. **Allow any ACL file to override masks** box checked:

   In this case, the masks defined in the protect directive are ignored by access control. The protection is taken entirely from the ACL file and the rules it contains. In this case, we say that the documents in the directory are protected in the following way:

   - All users in the validation list can GET or POST to admin1.htm. The GET and POST masks that we implemented on the protect directive are now irrelevant as the ACL is overriding those subdirectives.
   - Users Jack and Mike can GET or POST to admin2.htm and no one else.

The preceding examples are based on the premise that all the users have valid userids and passwords in the validation list referenced by the protection setup.

8.4.13 Adding an Access Control List

Let's see how to add an ACL file and review the rules we have just discussed. From Configuration and Administration Forms window, click on **Access Control Lists**. The Access Control Lists form is shown (Figure 128).

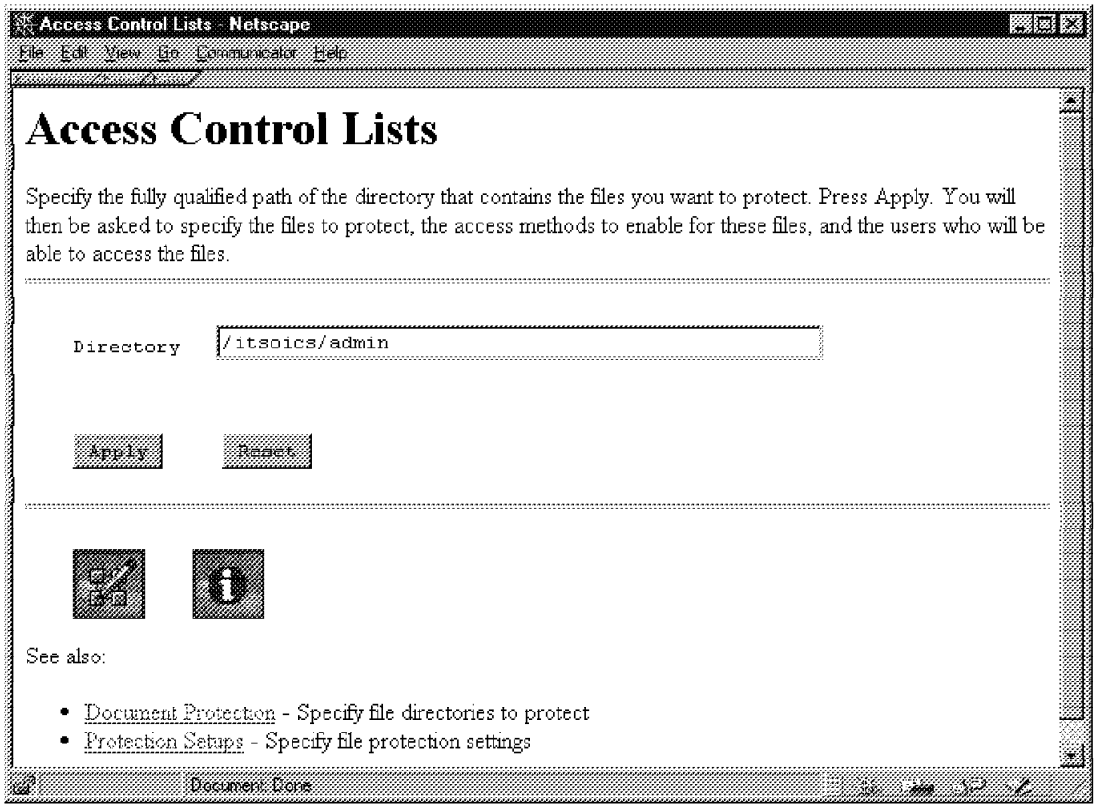


Figure 128. Access Control List - Specifying a Directory

Enter the directory that contains the files you want to protect and click **Apply**. The Access Control List form is shown (see Figure 129 on page 155).

For more information, click on the **Help** icon.

8.4.14 Specifying ACL Rules

Having specified the directory to contain the ACL list, we must now specify how the directory is to be protected.

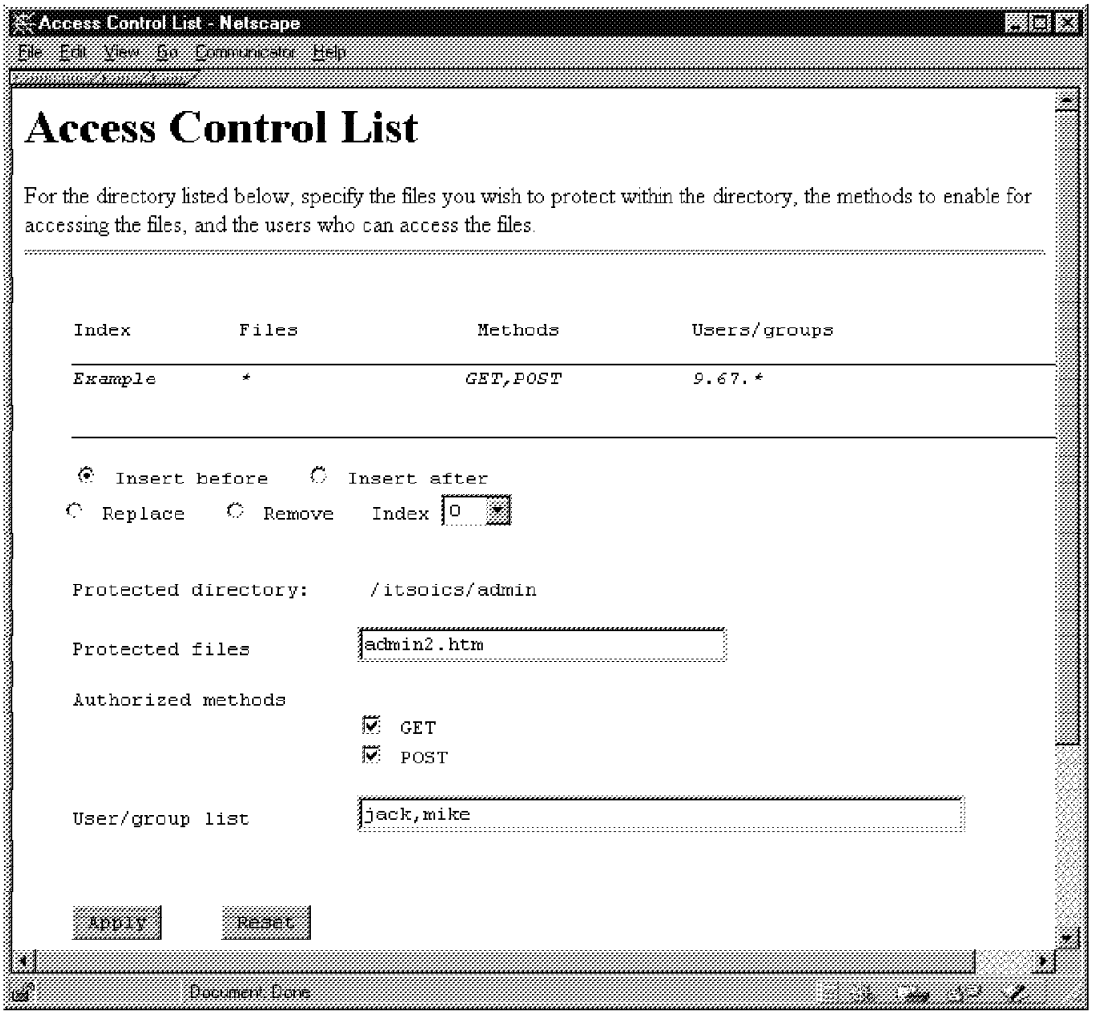


Figure 129. Specifying Files to Protect

In Figure 129, we have added a rule to protect the document admin2.htm such that only users Jack and Mike can use GET and POST methods. After completing this form, click on **Apply** to add the rules to the ACL file in the specified directory. If one does not already exist, it is created automatically for you.

Remember that there should be rules for all documents. For example, if you do not allow users to access image files such as GIFs, they are not served to the browser.

8.4.15 ACL Operation

It is possible, using Client Access or some other means of viewing text files on the AS/400 system, to look at the content of the Access Control List file.

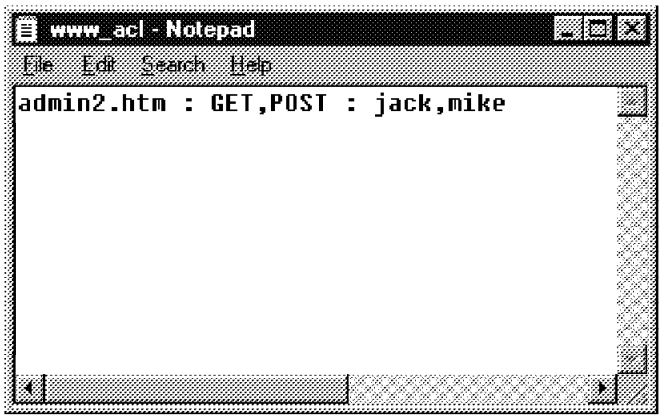


Figure 130. Access Control List www_acl in Directory /itsoics/admin

Figure 130 shows the contents of file www_acl that we just added to the /itsoics/admin directory as a result of the procedures shown in Figure 128 on page 154 and Figure 129 on page 155.

An attempt by someone (not authorized) to access the protected files results in an authorization failure as shown in Figure 98 on page 123.

Because of the default fail rule, any document in the directory not mentioned in the ACL file rules is not accessible and results in an error shown in Figure 131 on page 157.

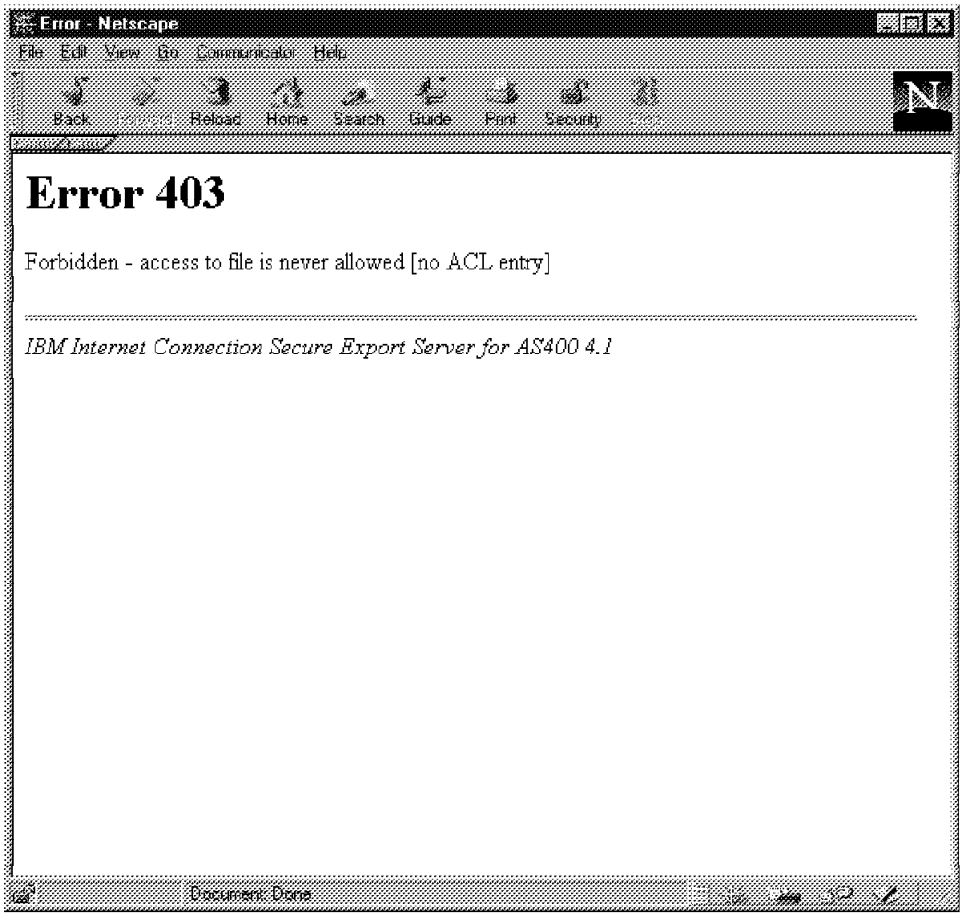


Figure 131. Access Control List Authorization Failure

Therefore, ensure that all the documents in a directory protected by an ACL are covered by a rule in the ACL.

8.4.16 How the Server Passes Requests

To ensure that access control is configured as efficiently and securely as possible, it is essential that you know how an incoming request is processed by access control.

The following description shows how the server processes a request that has already activated protection and been accepted by a Pass or Exec directive. The description assumes that all protection is defined in the protection setup (no ACL file exists on the protected directory).

Read over the description now to help you decide what type of protection you want to use. You may want to read it again in more detail after following the steps to create a protection setup:

1. Based on the HTTP method of the request, the server refers to the appropriate mask subdirective (GetMask, Mask, or PostMask) in the protection setup. The mask subdirective specifies valid user names, groups, or address templates.
2. If any items on the mask subdirective use only the address template protection, the server compares the address of the requester against the

address templates. Items that use only address template protection start with either @, Anybody@, Anyone@, or Anonymous@ followed by one or more address templates. Group names on the subdirective might also contain items that use only address template protection.

If there is a match, the server completes the request without prompting for a user name or password.

If there is not a match or no items use address template protection only, the process continues with the next step.

3. If any items on the mask subdirective are user names or group names, the server prompts the requester for a user name and password.
4. The server checks the user name sent by the requester against the valid user names. Valid user names are either the individual user names on the mask subdirective or user names defined as being part of a group file that is on the mask subdirective.

   If there is a match, the process continues with the next step.

   If there is no match, the process ends and the server returns a message to the requester saying that authorization failed.
5. If the user name sent by the requester is also associated with an address template, the server checks the address of the requester against the template. The mask subdirectives and group files use the at sign character (@) to associate user names or group names with address templates.

   If there is a match, the process continues with the next step.

   If there is no match, the process ends and the server returns a message to the requester saying that authorization failed.
6. The server checks the user name sent by the requester against the user names in the validation list object that the protection setup points to (or in the case of %%SYSTEM%% on the PasswdFile subdirective, against the AS/400 user profile database).

   If there is a match, the process continues with the next step.

   Note

   It is important to note that the validation list must contain an entry for the user name that the requester sends to the server. You make up the user names that are in the validation list. The names themselves do not have any relation to the addresses of the requesters.

   If there is no match, the process ends and the server returns a message to the requester saying that authorization failed.
7. The server checks the password sent by the requester against the password defined for the user name in the validation list. Each user name in the validation list has one valid password.

   If there is a match, the server completes the request.

   If there is no match, the process ends and the server returns a message to the requester saying that authorization failed.

Chapter 9. Establishing a Secure Connection

In this chapter, we look at the steps required to implement a secure server on the AS/400 system.

The AS/400 system can act as a secure server by installing and configuring Internet Connection Secure Server for AS/400 (ICSS for AS/400). While you may act as a certifying authority (CA) utilizing Internet Connection Secure Server for AS/400, the normal way is to request a certificate from a CA. In this chapter, we look at:

- How to request and receive a certificate from a CA and activate SSL
- How to configure an AS/400 system to act as a CA in an intranet
- How to process certificates for servers within the intranet

At the end of the chapter, there are several step-by-step procedures for SSL configuration as well as a planning form. These are *not* accompanied by descriptions or explanations, but are intended as fast-paths to configuration. If you are not familiar with SSL, HTTPS, and Internet Connection Secure Server for AS/400, we recommend that you read Chapter 4, “AS/400 Internet Connection Secure Server” on page 15, which explains how SSL and HTTPS work. Also remember that the help text pages of the ICSS for AS/400 are there to be used.

ICSS for AS/400 configuration is done from the Security forms of the Configuration and Administration Forms (see Figure 132 on page 160). This chapter requires that you already have a working serving instance.

Remember

For the changes you make to the server to take effect, the server instance must be *stopped* and then started again.

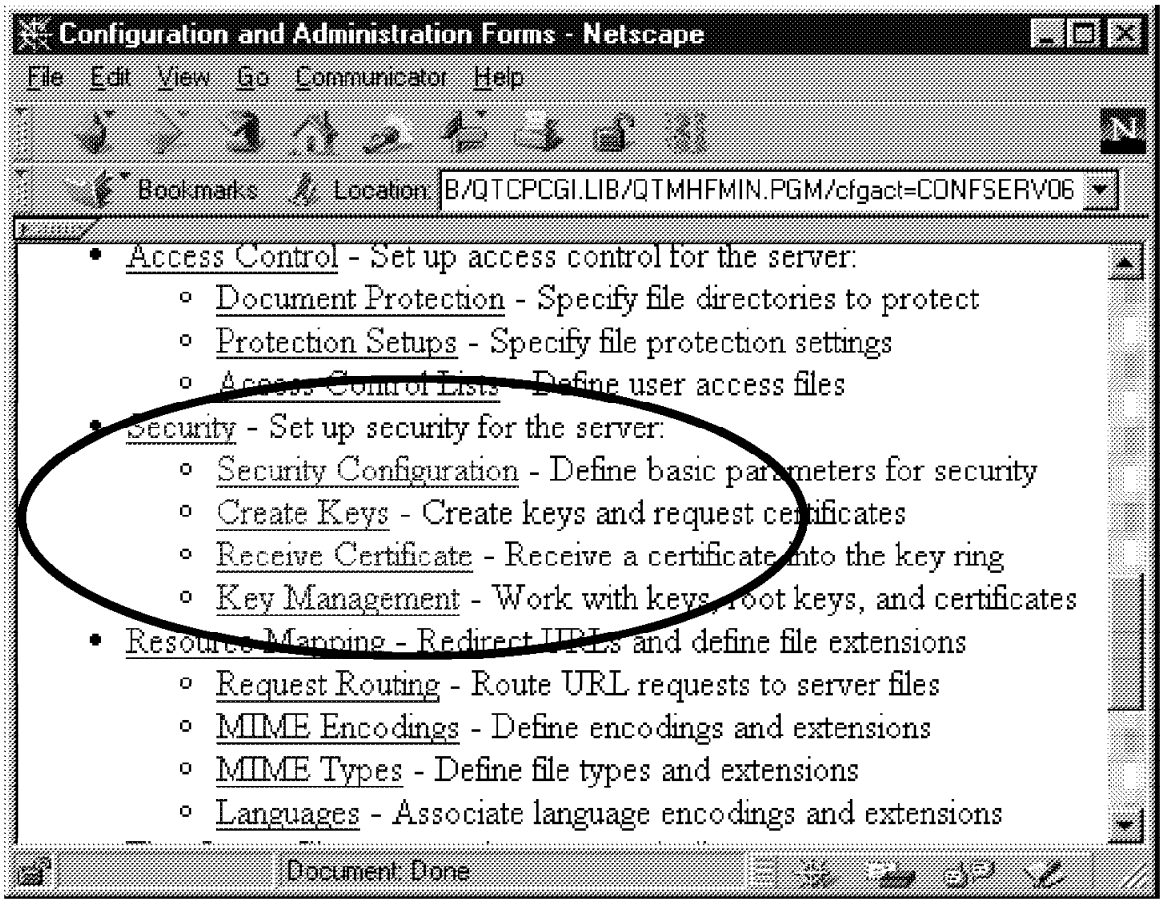


Figure 132. Configuration and Administration Form: Security

9.1 Setting Up SSL

This section guides you through the basic SSL setup. You must specify a port for SSL (typically 443); you must have a private-public key pair and a certificate, and finally, you have to enable SSL.

The procedures described in this section are intended primarily for the situation where you want to enable SSL on the server. Some of the steps in the procedures are identical or similar to the procedures in Section 9.3, “Acting as a Certificate Authority” on page 194.

9.1.1 Allow SSL Connections

From the Configuration and Administration Form, select **Security Configuration**. This takes you to the general security options form for HTTP and SSL. This form allows you to open for SSL and set the SSL port as well as remove or set key rings. In this first step, we open this server instance to allow SSL connections.

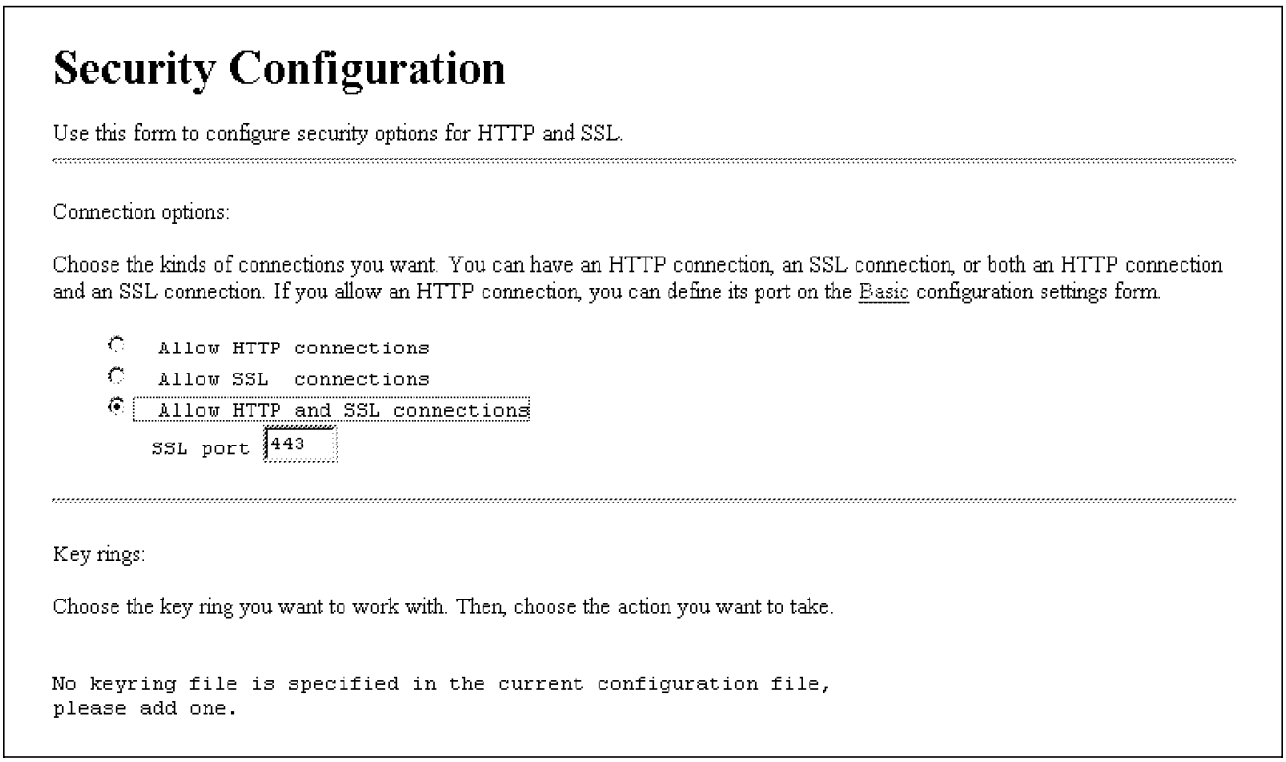


Figure 133. Allow SSL and Specify Port

Select **Allow HTTP and SSL connections** and specify a port for the secure connections. If you are configuring the server for Internet serving, use port 443. For intranet serving, however, you may use any port above 1024. Note that at this time, no key ring file has been specified. The SSL port number can be overridden by the -sslport option in the startup values of the STRTCPSVR command (see 6.11, “Start TCP/IP Server and End TCP/IP Server Commands” on page 102) or through instance parameters (see Figure 35 on page 56). Press the **Apply** button and check the *Confirmation form*.

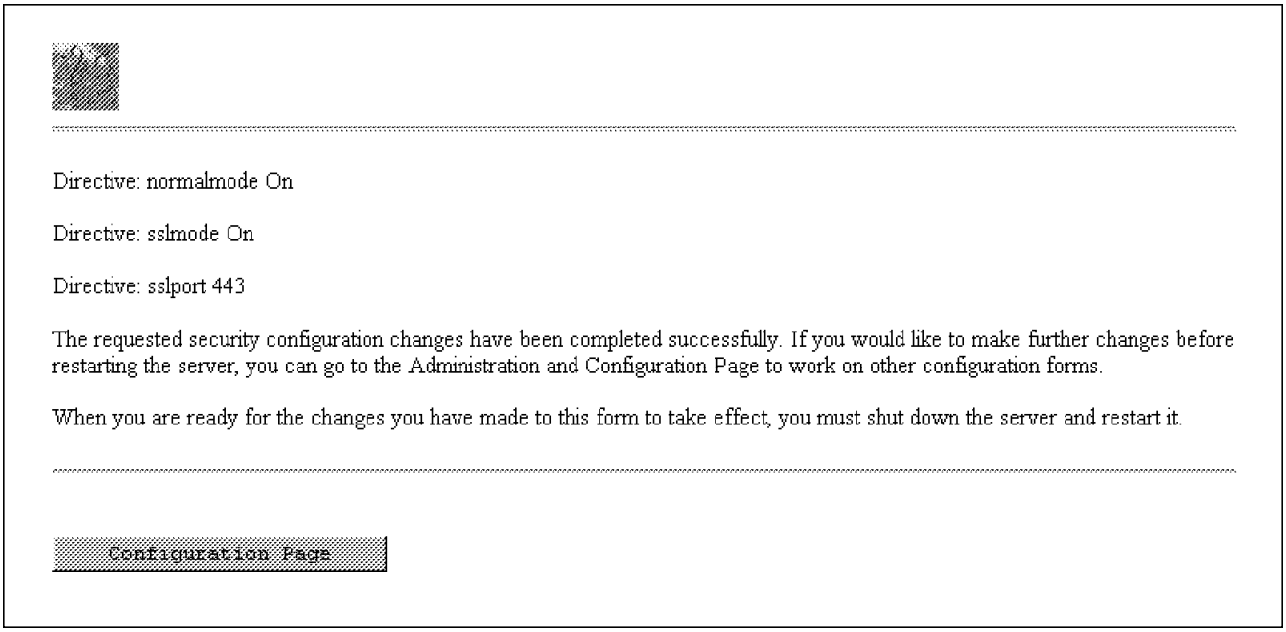


Figure 134. Confirmation Form

Press the **Configuration Page** button to return to the Configuration and Administration Form.

9.1.2 Request a Server Certificate

Now that you have opened for SSL connections, you must ″create a key and request a certificate″. Select **Create Keys** under Security from the Configuration and Administration Form.

Figure 135. Create Key and Request Certificate Form

Select either **VeriSign** or **Other**. If you are setting up a secure Internet server, you must request a certificate from VeriSign or another CA. This other CA may be some other external CA (for example, RSA) or, in an intranet configuration, it may be someone within the organization acting as a CA.

The next page is virtually identical for both *VeriSign* and *Other*. The main differences are in the headings (“VeriSign Secure Server Certificate” or “Other Certificate”) and in some of the form default values.

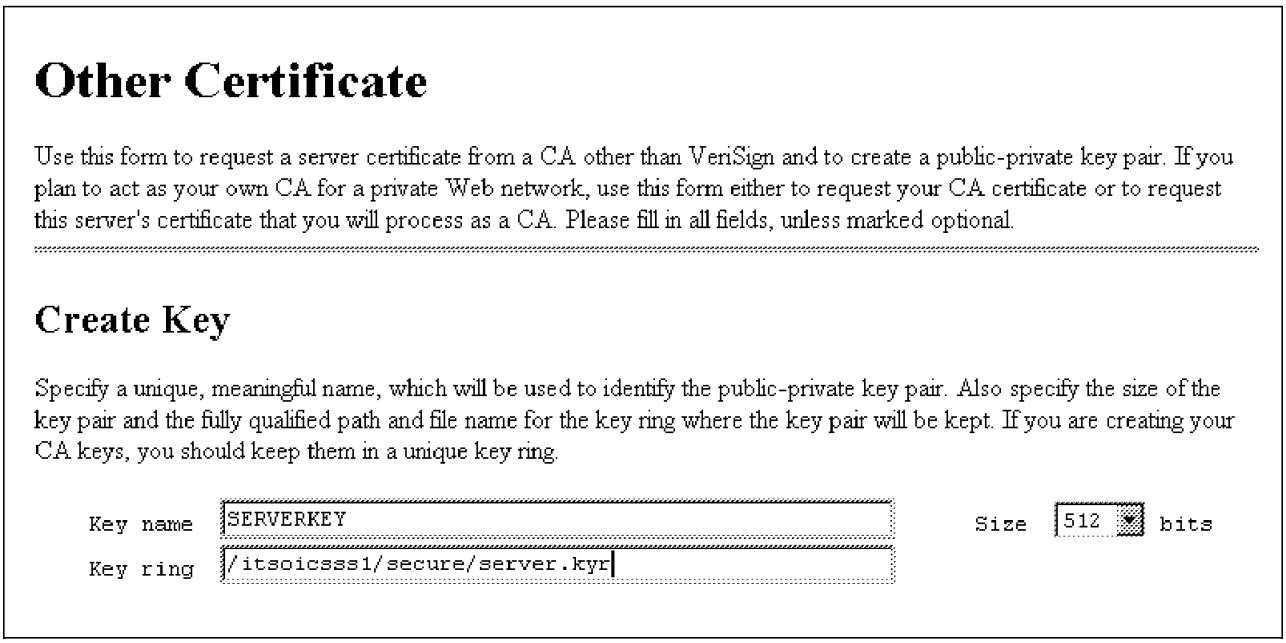


Figure 136. Specify Key Name and Path

For the “Create Key” values, specify:

- The name of the key:

  This should be a meaningful and unique name such as “SERVERKEY” or similar. It should clearly describe the usage of the key. It is also wise to avoid special characters in the key name.

- The maximum key size is dependent upon the licensed program (LP) you have installed - 5769-NC1 (U.S. and Canada) or 5769-NCE (International). The more bits you specify for the key pair, the more secure the communications are.

- The key ring is where the key pair is stored. As with the key name, this should be a meaningful name.

Note

The key file is created automatically but the directory must exist. After completion of the “Create Key and Request Certificate” form, a new key ring file can be seen as a *.KYR file within the specified directory.

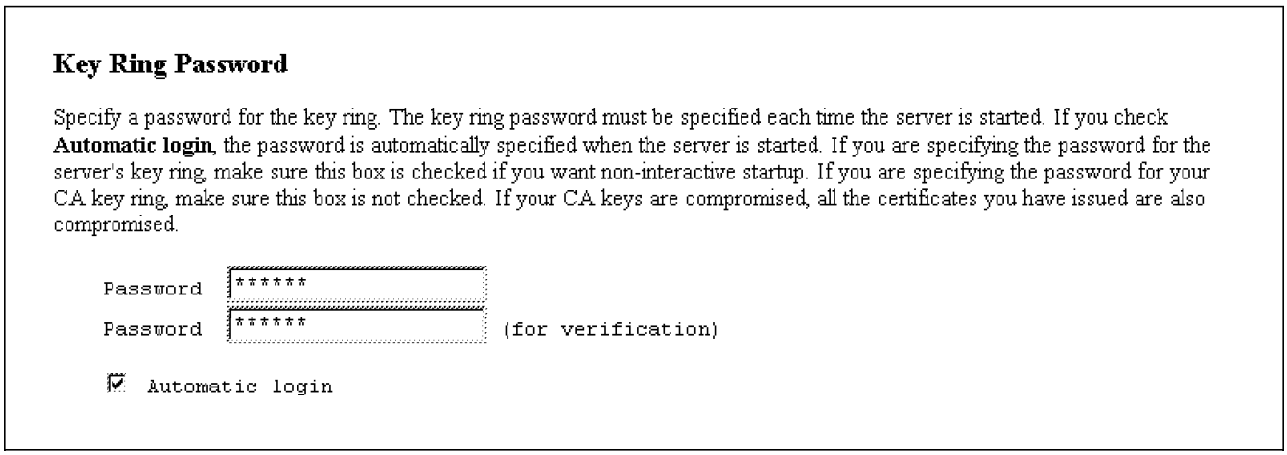


Figure 137. Specify Key Password

Specify a password for a *new* key ring file or the valid password for an existing key ring file. If you are creating the first key for this server, the key ring file is also new. Observe the warnings in the text of the password fields. Follow these simple rules:

- As this password protects the private key, that key may be compromised if the key ring password is compromised.
- The key ring password is case sensitive so be careful with upper and lower case characters within the password.
- All common password rules also apply to key ring passwords.

  Note: AS/400 system values do *not* control key ring passwords.
- Check the **Automatic Login** box to specify that the password is to be automatically specified when the server is started. If you do not check this box, the server cannot be started.

You must type the password twice to ensure that you have typed it correctly. The password is not visible while you type it. A password file is created in the same directory as the key ring file; it is encrypted, has an .STH extension, and should be sufficiently protected.

The next step is to fill out the certificate request. It consist of three parts: Distinguished Name, e-mail options, and a save to file parameter.

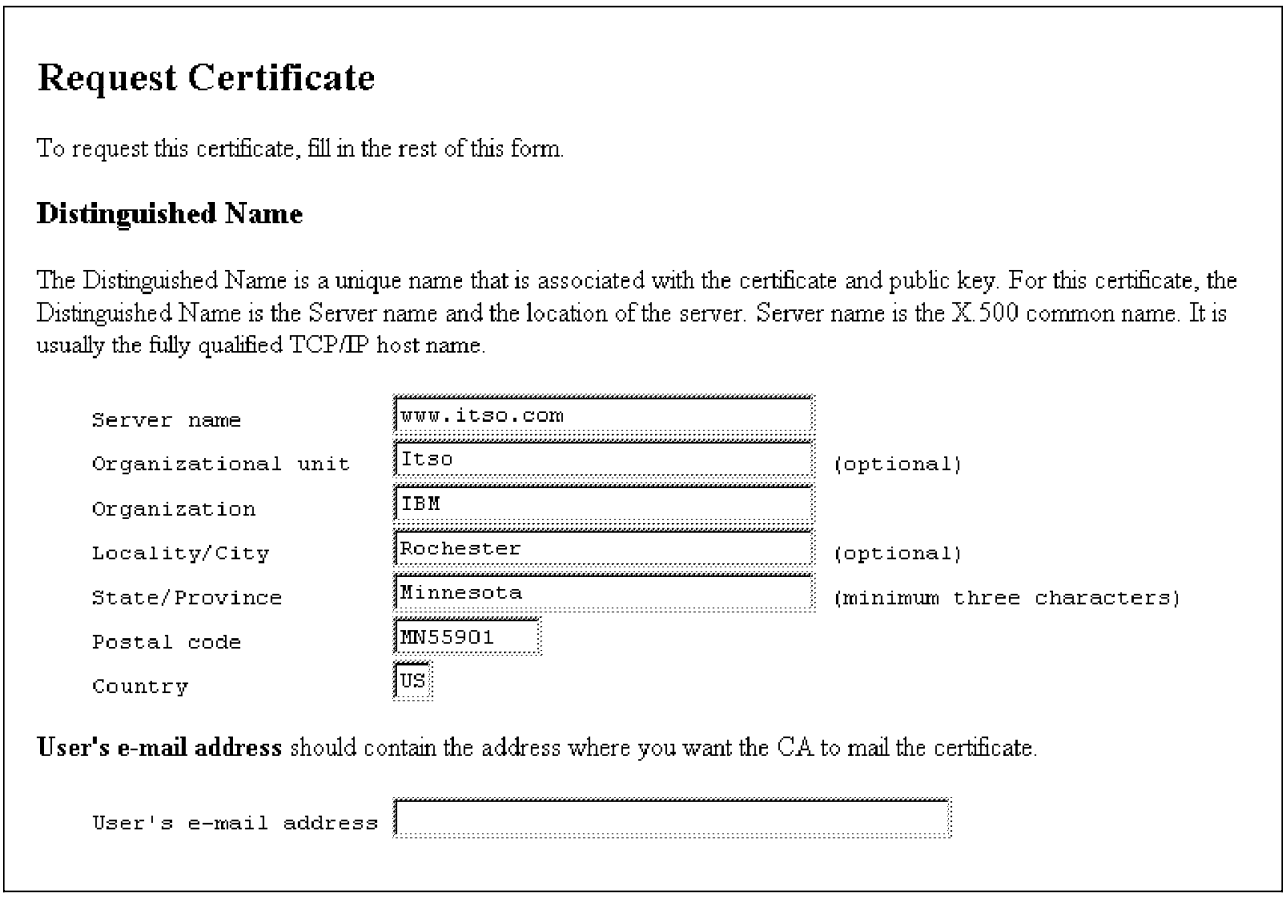


Figure 138. Request Certificate, Specify Distinguished Name

The distinguished name of a server is the name that uniquely identifies this server:

- **Server name**: Type in the fully-qualified host name, usually the X.500 common name of the server. It is normally in the form *servername.domain.com*. You can find it as part of the TCP/IP configuration (CFGTCP + option 12).
- **Organizational unit**: This information is optional but may be used to specify the division, section, or company within an enterprise.
- **Organization**: The official name of the organization. However, if the company already has an account with, for example, VeriSign, this field should match the name of that account.
- **Location/City**: This is an optional field but should be used to specify the server's geographical whereabouts.
- **State/Province**: The general geographical area in which the server is located (this information must be at least three characters long).
- **Postal Code**: Type the postal or zip code for the server's geographical whereabouts.
- **Country**: This should be your country's two-character code. The ISO-3166 standard for the representation of the name of your country should be used.

If you want the CA to e-mail you the processed certificate, specify your e-mail address in the **User's e-mail address** field.

The **Mail to** option enables you to e-mail the request to your CA. Refer to the CA's instructions to determine how to mail the certificate request. Note that some e-mail programs alter files and should not be used to send requests. Also, if you are behind a firewall, verify with the system administrator what you need to do to electronically mail the request.

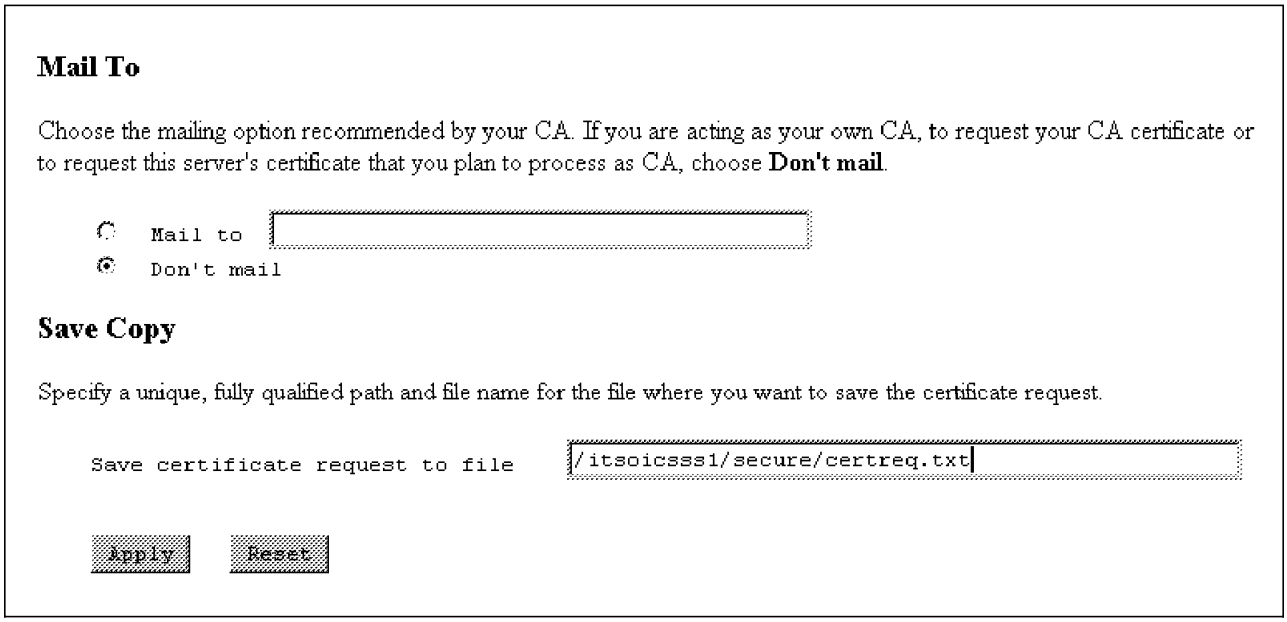


Figure 139. Request Certificate, Specify Mail, and Save Parms

Save certificate request to file: This allows you to save the request in a file. This file *may* be used to snail-mail the request to your CA. You must specify a fully-qualified path for the new request file. We recommend that you give the file a name that indicates its nature as a certificate request file.

Press the **Apply** button to process the Key and Certificate Request form.

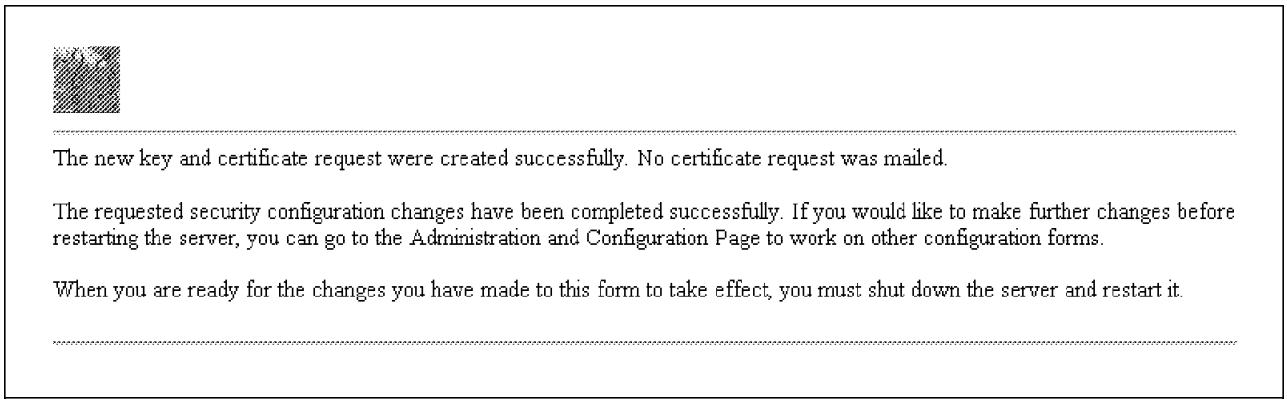


Figure 140. Create Key and Request Certificate Confirmation

You should receive a confirmation page stating that the *new key* and *certificate request* were created. The confirmation also tells you if the request was mailed or not.

As a result of the request, the server has:

- Created (or updated) the *KeyRing*.KYR file.
- Stored the *key pair* in an encrypted format in the key ring.
- Added the default trusted roots (VeriSign, RSA, Netscape) to the key ring. (These are shipped with ICSS for AS/400.)
- Added a *Key Ring*.STH password file and encrypted the key ring password into that file. This is a result of checking the Automatic login box.
- Created the certificate request and stored a copy in the file you specified.

Press the **Configuration Page** button to return to the Configuration and Administration Form.

If you browse the directories at this stage, you see that the new files have been created (or the old key ring file has been updated).

9.1.3 Receive a Server Certificate

When your CA responds to your request and returns your processed certificate, it must be received into the key ring. Before you can receive a certificate into a key ring, you must add the key ring as the *current key ring* for the server. The procedure required to receive a certificate into a key ring may also include receiving a CA key from your intranet CA.

9.1.3.1 Add the Key Ring File as the Current Key Ring

To activate the Key Ring file, add it as the *Current Key Ring*. This is done from the Security Configuration form. If you do not know if the key ring file you are working with is the current key ring, look at the Key Management form first. To add the key ring file as the current key ring, select the **Add key ring** button and type the fully-qualified file name of the key ring file as the new current key ring file. If the key ring file is listed in the *Key rings* box, select it and select **Set selected key ring as current key ring**.

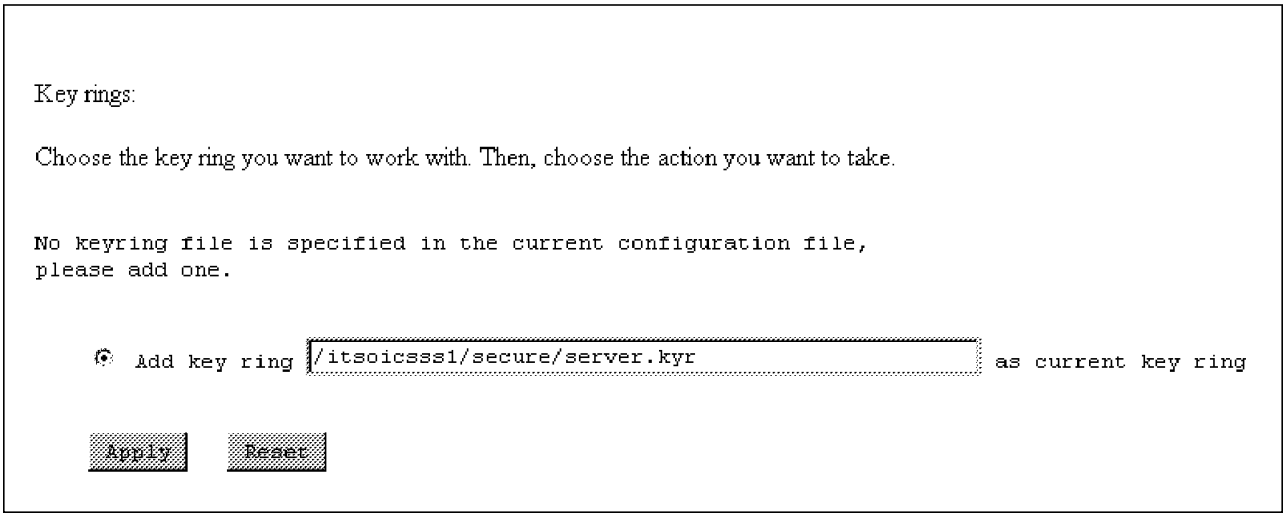


Figure 141. Add Key Ring

After selecting the correct button and if required, typing the file path, press the **Apply** button. The confirmation form should have the new key file specified in the *key file* directive.

Figure 142. Basic Security Configuration, Confirmation

Return by pushing the *Configuration Page* button. If your CA is an external organization such as VeriSign, you may now proceed to Section 9.1.3.3, “Receive Certificate from a CA (External or Internal)” on page 172.

9.1.3.2 Receive CA Certificate

This step is required if you receive a certificate from an intranet CA, or if your CA is not one of the default trusted roots shipped with ICSS for AS/400. Be *careful* if your CA:

- Is an external institution.
- The CA root is *not* one of the default trusted roots.

From the Configuration and Administration Form, select the **Receive Certificate** link. You are prompted with the Receive Certificate form.

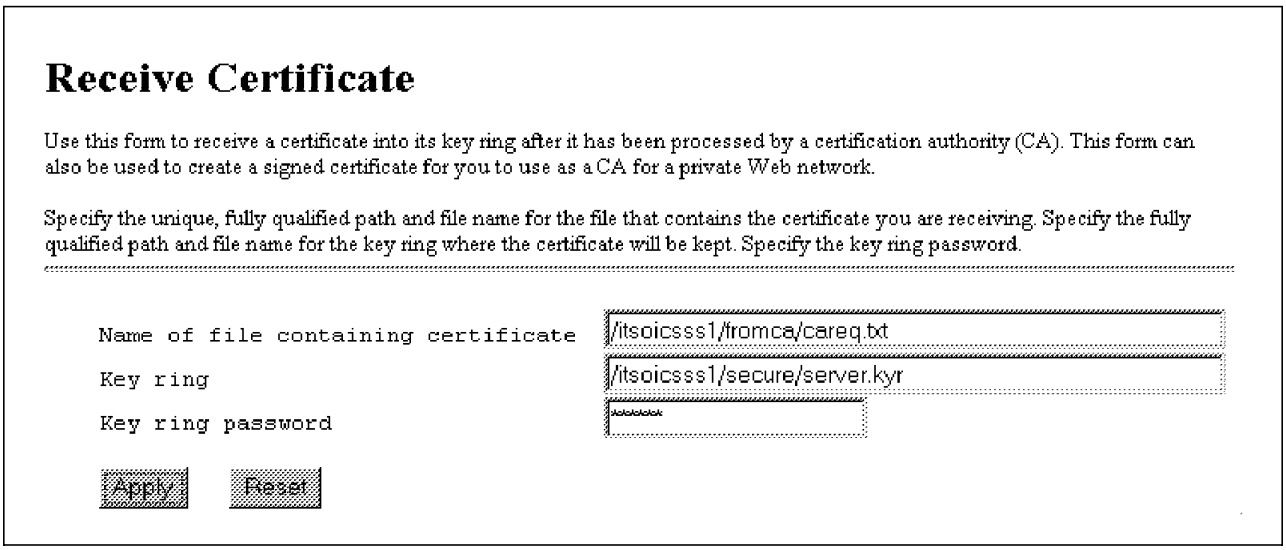


Figure 143. Receive CA Certificate, Path, and Password

Type in the fully-qualified path and **Name of file containing** *CA* **certificate** and the fully-qualified path and name of the **Key ring** file. Enter the **Key ring password** and press **Apply**.

You see a confirmation page, *Certificate successfully received*. Press the **Configuration Page** button to return to the Configuration and Administration Form.

From the Configuration and Administration Form, select **Key Management** under Security.

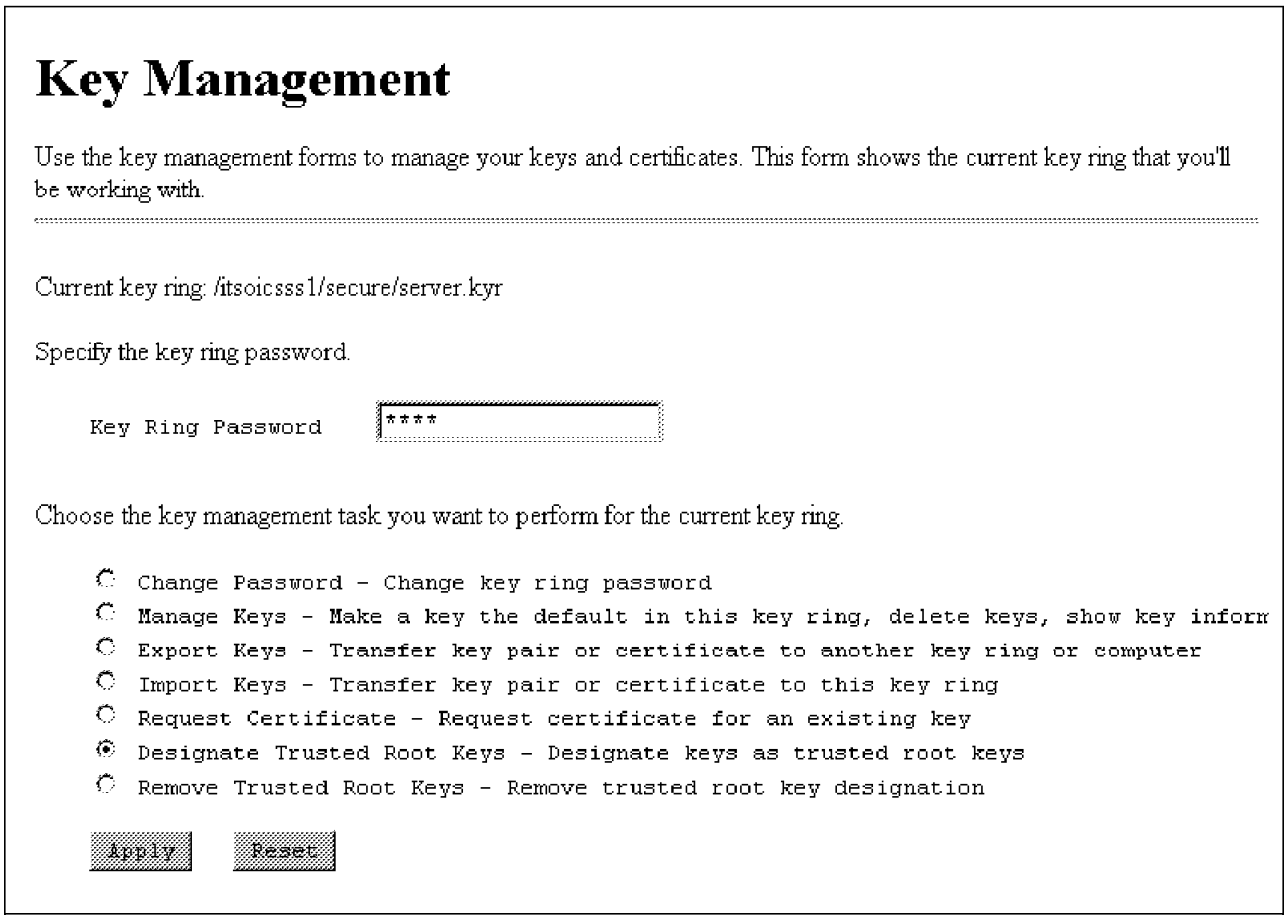


Figure 144. Key Management, Designate Trusted Root Key

On the Key Management page, enter the **Key Ring Password**, select the **Designate Trusted Root Keys** option and press the **Apply** button.

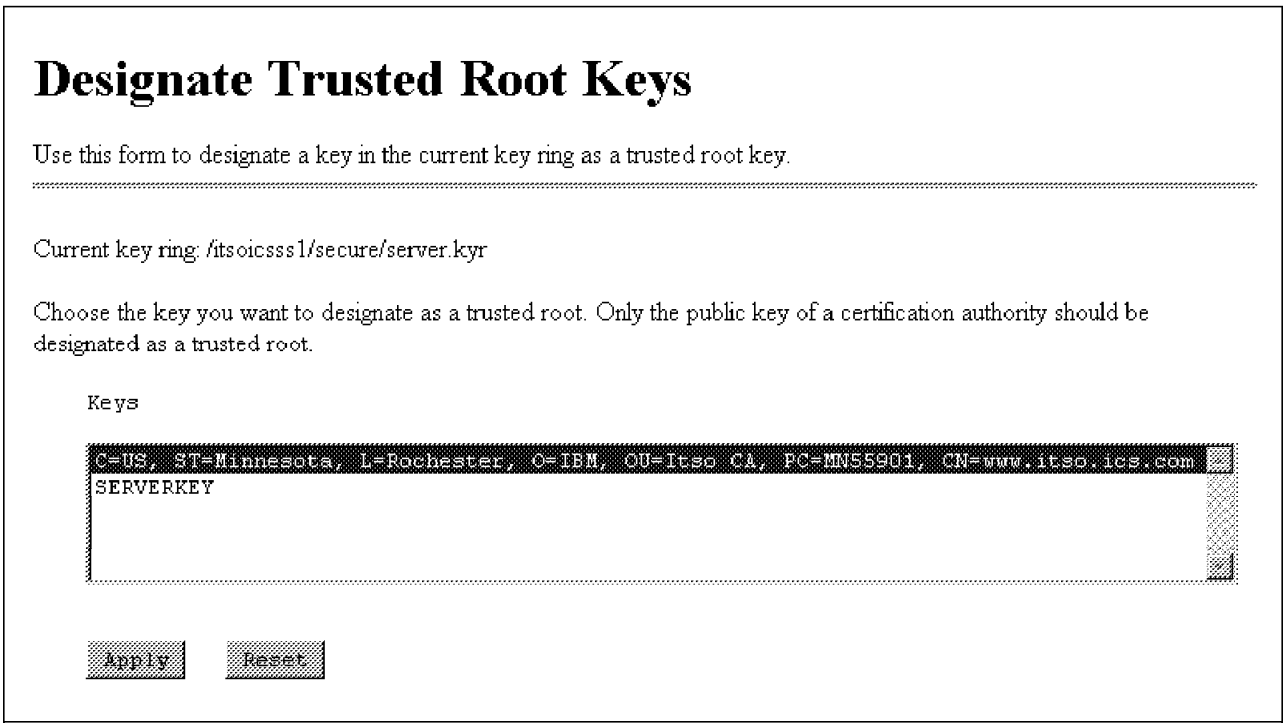


Figure 145. Designate the CA as a Trusted Root

Select the CA key in the **Keys** box and press **Apply**.

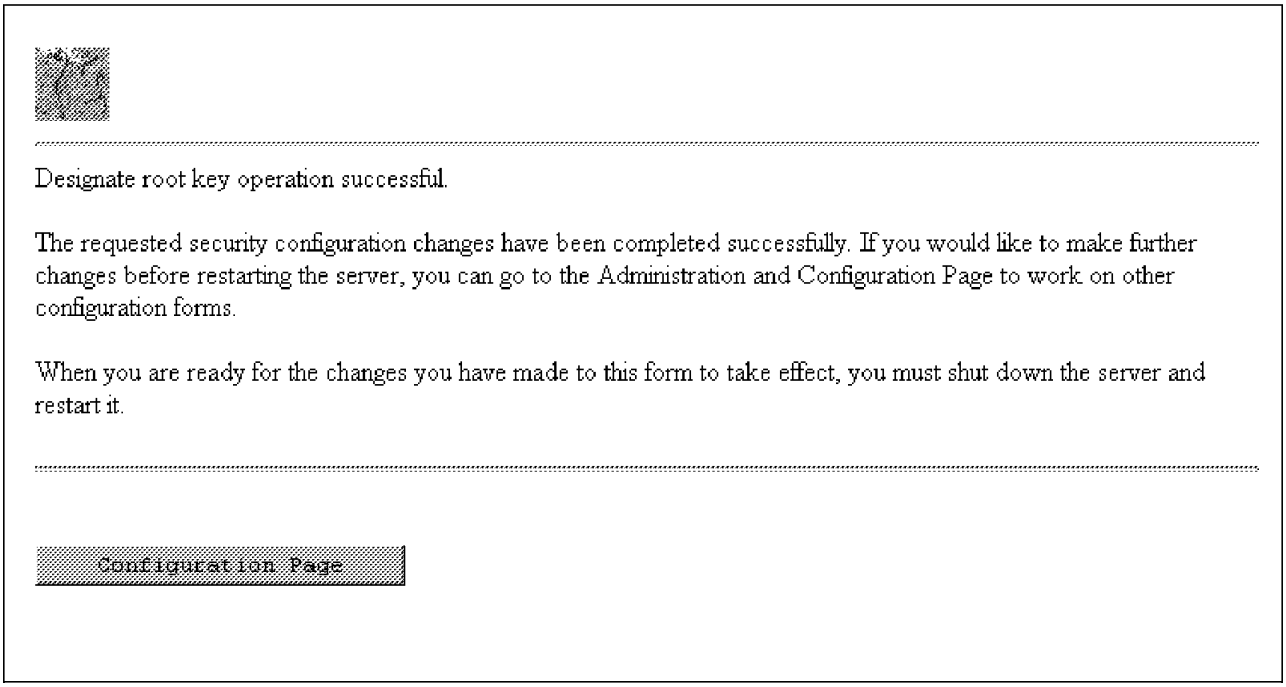


Figure 146. Designate Trusted Root, Confirmation

The *Confirmation page* indicates a successful *Designate trusted root* operation. Press the **Configuration Page** button to return.

Now that you have received the CA's certificate and designated it a trusted root, you are ready to receive the server certificate from the CA.

9.1.3.3 Receive Certificate from a CA (External or Internal)

The following procedure is fairly straight-forward. It is the same whether you are receiving an externally processed certificate or a certificate from an intranet CA. You should have the details of the key ring, key ring password, and the file containing the certificate before you start receiving the certificate.

From the Configuration and Administration Form, select the **Receive Certificate** link. You are prompted with the Receive Certificate form.

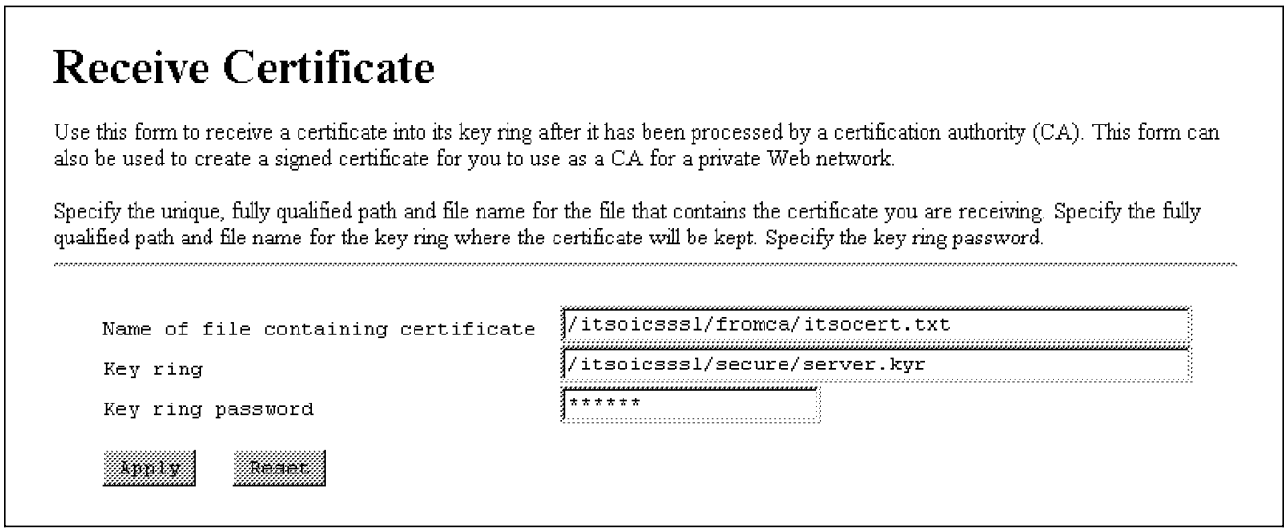


Figure 147. Receive Certificate, Path, and Password

Type in the fully-qualified path and **Name of file containing** *Server* **certificate** and the fully-qualified path and name of the **Key ring file**. Enter the **Key ring password** and press **Apply**.

You see a confirmation page, *Certificate successfully received*. Press the **Configuration Page** button to return to the Configuration and Administration Form.

9.1.3.4 Verify Certificate and Check Current Default Key

To look at the newly received certificate, select **Key Management** from the Configuration and Administration Form. Enter the **Key ring password**, choose **Manage Keys**, and press the **Apply** button.

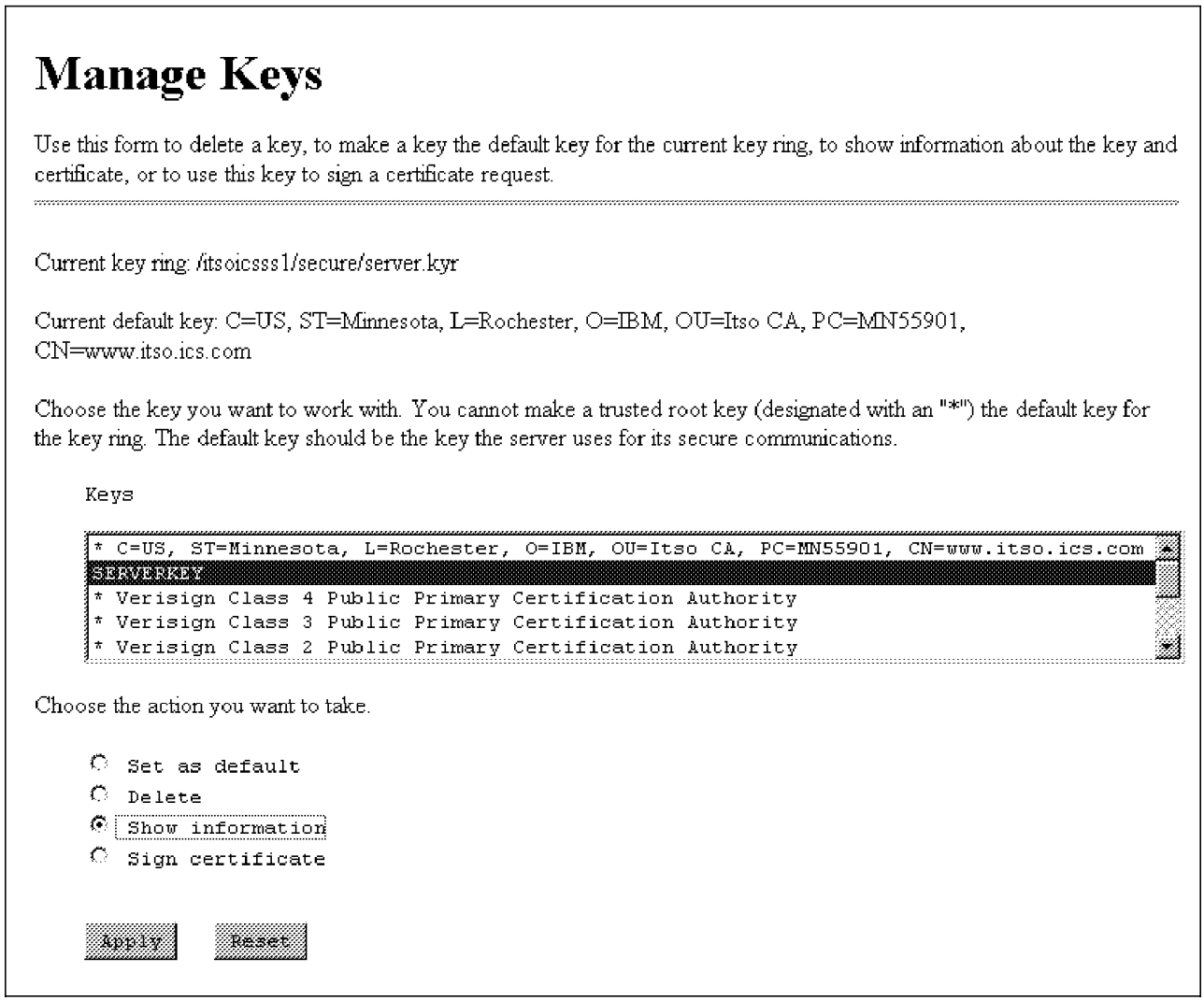


Figure 148. Receive Certificate, Manage Keys

From the Manage Keys page, select the **server key**, choose **Show information** and press the **Apply** button. The Key and Certificate Information page is displayed; it has four groups of information:

Key and Certificate Information

Selected Key: SERVERKEY

- 512 bits long
- Has a private key
- Is not a trusted root
- Has a signed certificate

Issued to:

- Common name: www.itso.com
- Organizational unit: Itso
- Organization: IBM
- Locality: Rochester
- State/Province: Minnesota
- Zipcode: MN55901
- Country: US

Figure 149. Receive Certificate, Key, and Issued to Info

- The first group contains the key information (length, type, and so on).
- Group number two contains information about the server.
- Group number three contains the information about the CA.
- The last group is for the certificate itself, its number, and duration.

Issued by:

- Common name: www.itso.ics.com
- Organizational unit: Itso CA
- Organization: IBM
- Country: US

Certificate:

- Serial number: 34FECFCB
- Valid from 03/05/98 to 03/05/99

Figure 150. Receive Certificate, Issued By, and Certificate Info

9.1.3.5 Set Server Key as Default Key

Before you can verify secure connections, you must ensure that the server key is the *Current default key*.

Select **Key Management**. Next type the **Key ring password**, check the **Manage Keys** button, and press **Apply**. The Manage Keys form is displayed.

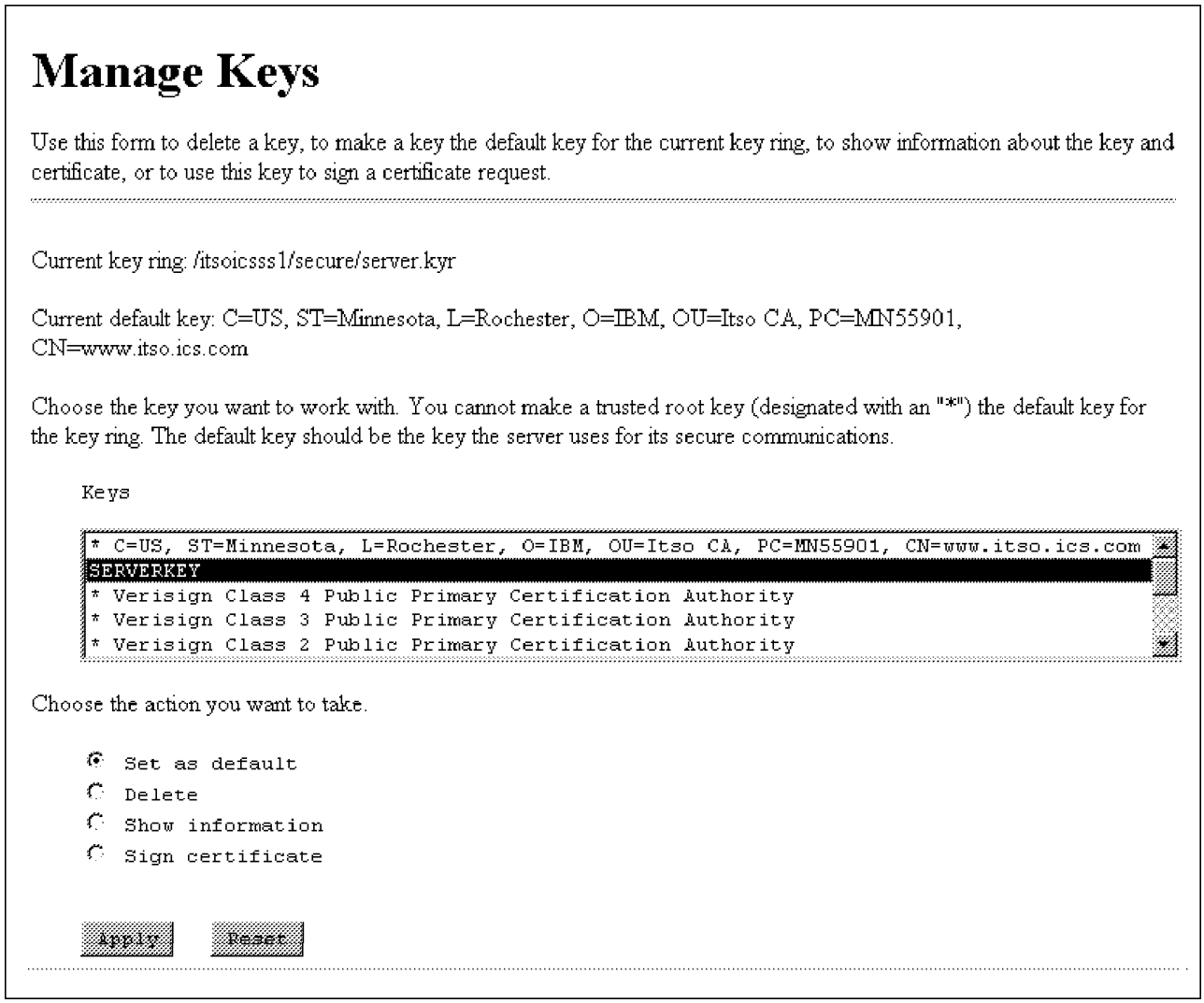


Figure 151. Set the Server Key as the Current Default

If the Server key is not the *Current default key*, you must select it from the **Keys** box, select the **Set as default** button, and press the **Apply** button.

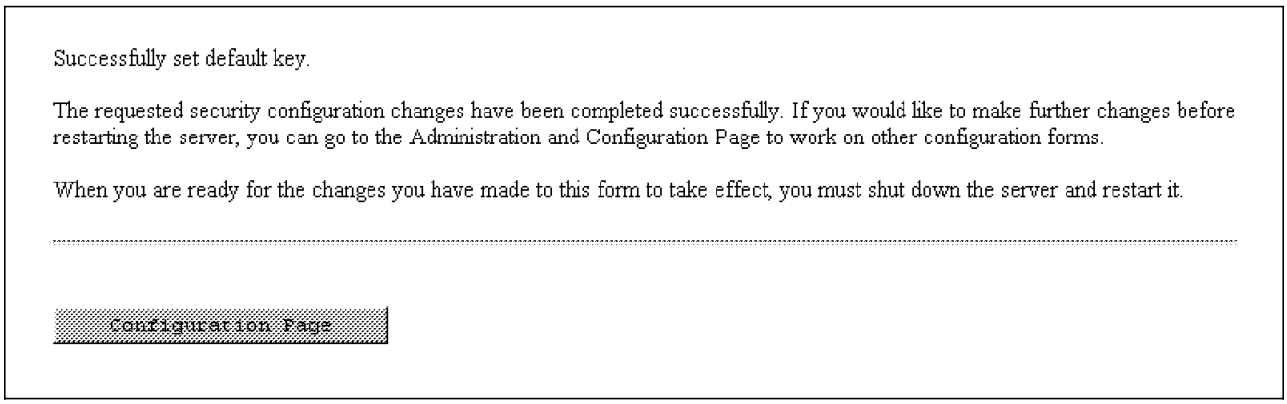


Figure 152. Confirm New Current Default Key

Press the **Configuration Page** button to return.

9.1.3.6 New Directives in the HTTP Config File

If you look at the HTTP configuration file, WRKHTTPCFG CFG(myserver), you can see these directives.

```
                HTTP Configuration File

          normalmode       on
          sslmode       on
          sslport       443
          keyfile /itsoicsss1/secure/server.kyr
```

Figure 153. SSL Directives Added to HTTP Configuration File

Note that the “normalmode” directive may not be positioned together with the other SSL directives.

9.1.4 Verify Secure Document Serving

Now that we have received the new certificate, it is time to verify the SSL connection. This is done by simply accessing the server by the specified SSL port, but before you can do that, you must:

- Stop and start your server.
- Close your browser session and restart it.
- Ensure that your browser supports SSL. Netscape Navigator 3.x and 4.x support SSL as does Microsoft Internet Explorer 2.x, 3.x and 4.x.

If you now access the server through the dedicated SSL port using HTTPS, you are prompted with some, if not all, of the following windows (depending upon your browser, the pages may look different but the contents and information given in them are more or less the same).

The windows in the following pages are shown if the CA that generated the certificate *is not* in the browsers list of trusted roots (CAs that it trusts).

Figure 154. Start a Secure Session, Warning: Unknown CA

The first page is simply a warning that you are about to enter an encrypted site and that the browser does not recognize the CA. If you have a VeriSign or similar certificate, the browser should recognize the CA.

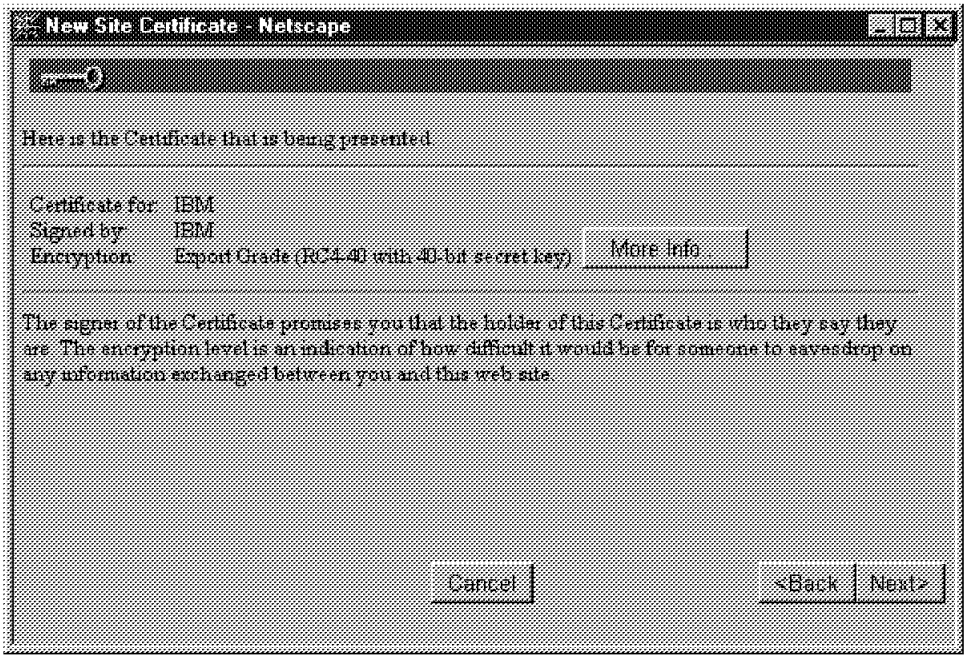


Figure 155. Start a Secure Session, Certificate Info

The second page in this example is certificate information. If you choose **More info..**, your browser will display details about the certificate.

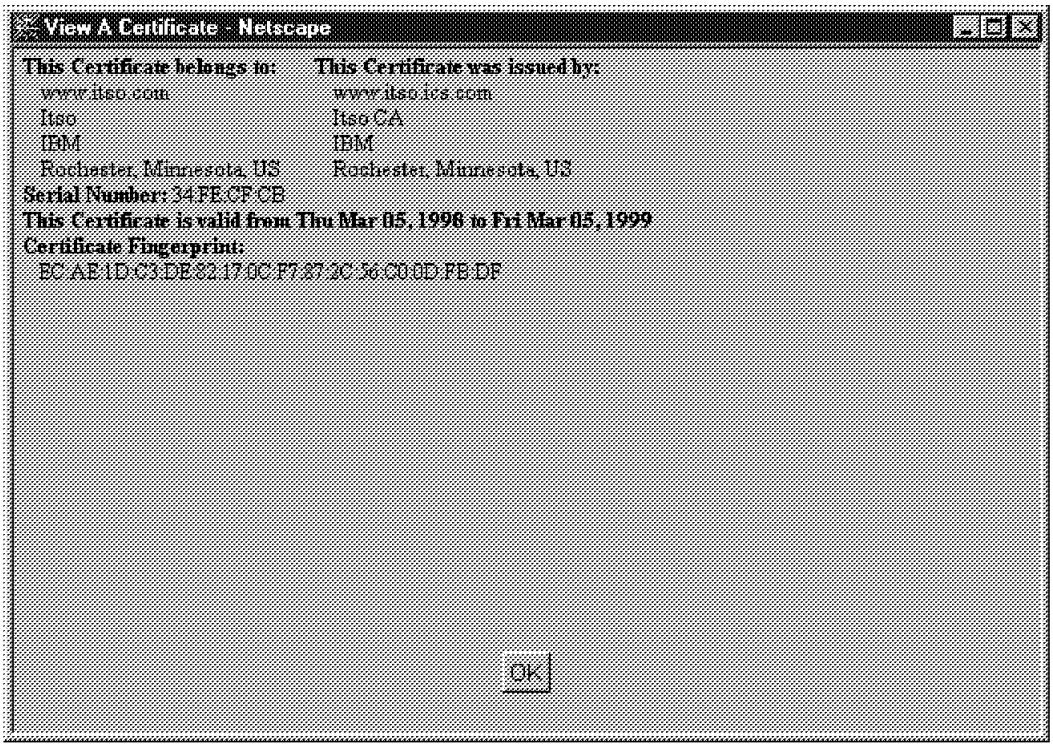


Figure 156. Start a Secure Session, Certificate Details

The next page is a challenge to you; are you willing to accept the certificate?

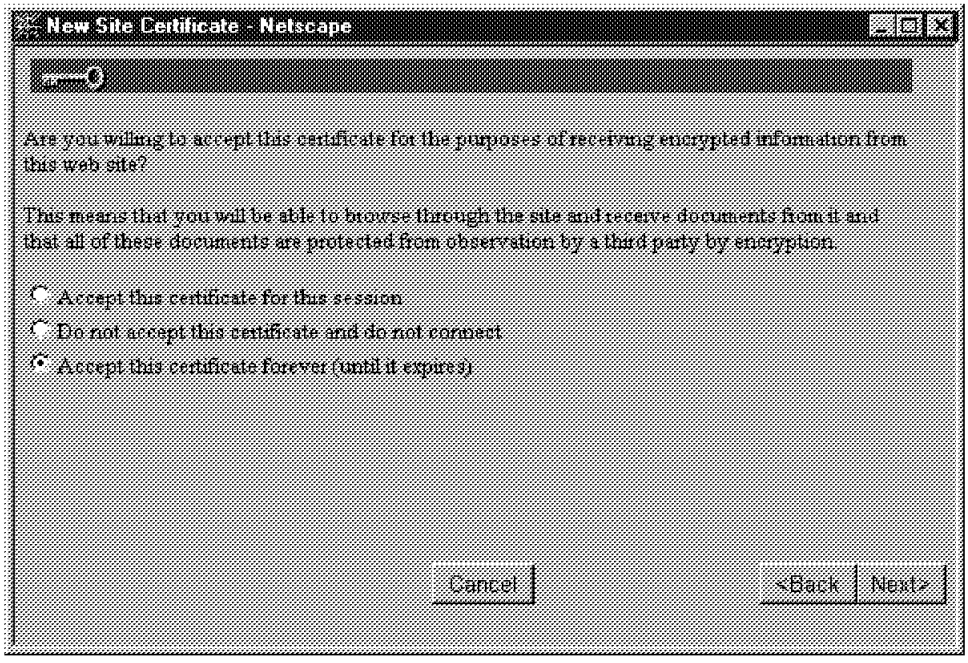


Figure 157. Start a Secure Session, Certificate Acceptance

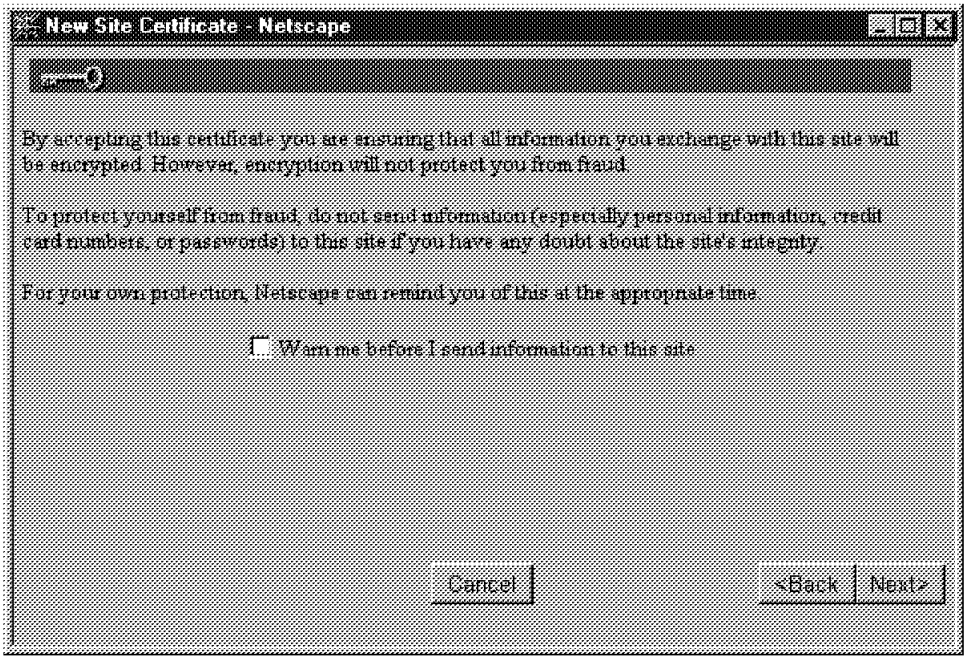


Figure 158. Start a Secure Session, Warning 2

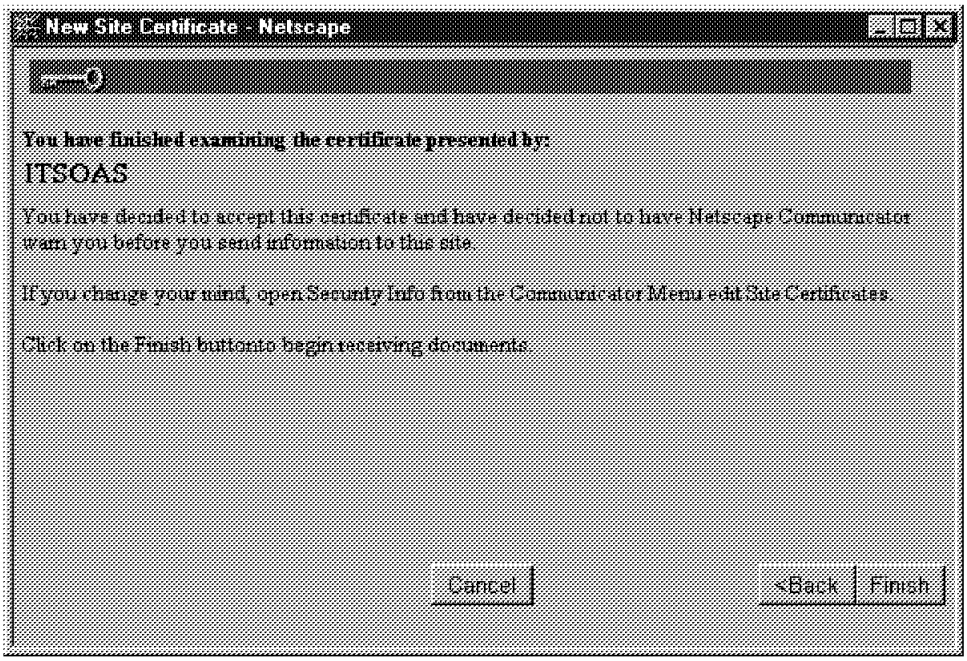


Figure 159. Start a Secure Session, Certificate Setup OK

At this stage, the server and browser have negotiated the SSL encryption suite, the session keys, and are communicating over an encrypted link. Somewhere in your browser, you see an icon indicating security. This icon may be a padlock (Netscape Navigator 4.x and Microsoft Internet Explorer), a key (Netscape Navigator 2.x and 3.x), or another symbol depending on your browser.

Netscape Navigator 4.0 **Microsoft Internet Explorer 4.0**

Figure 160. Secure Session Indication Icon

From your secure session, you can display the certificate information. From Netscape Navigator 4.0, select Security from the Navigation Toolbar and select **Certificates, Web sites**. Select the server certificate and press the **Edit** button. A page similar to Figure 161 is shown:

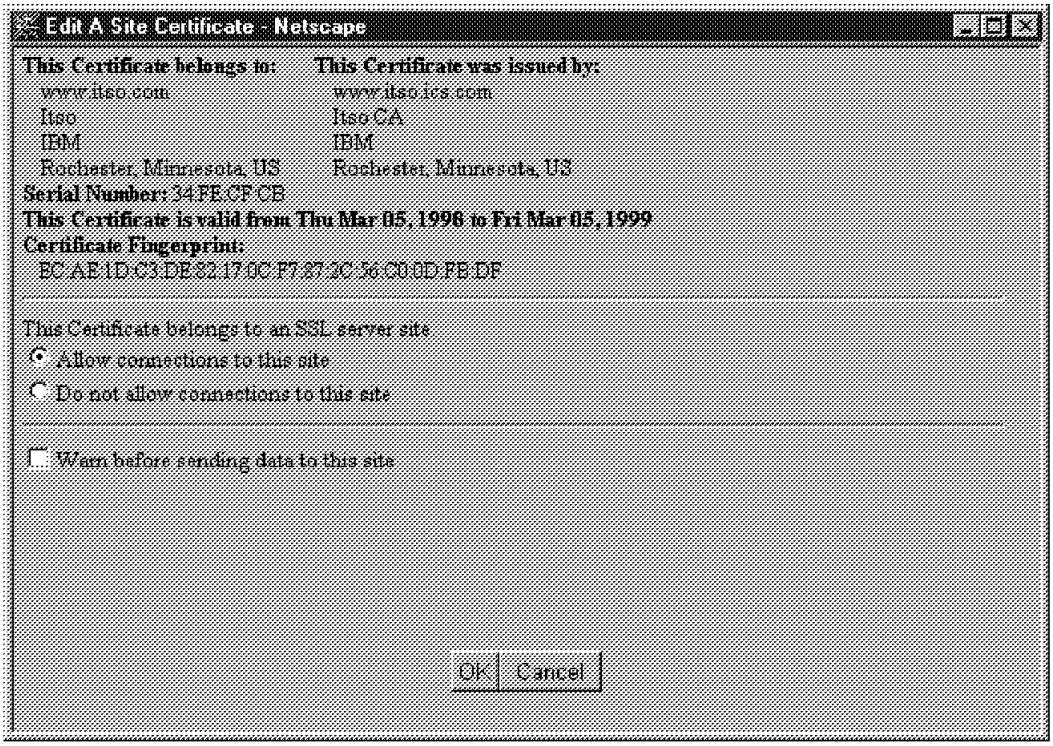


Figure 161. Certificate and Secure Session Information, Netscape Navigator

From Microsoft Internet Explorer, select **File**, then **Properties**, and press the **Certificates** button. On the “Properties” page, you may click on each of the **Field** descriptions to view the detailed information.

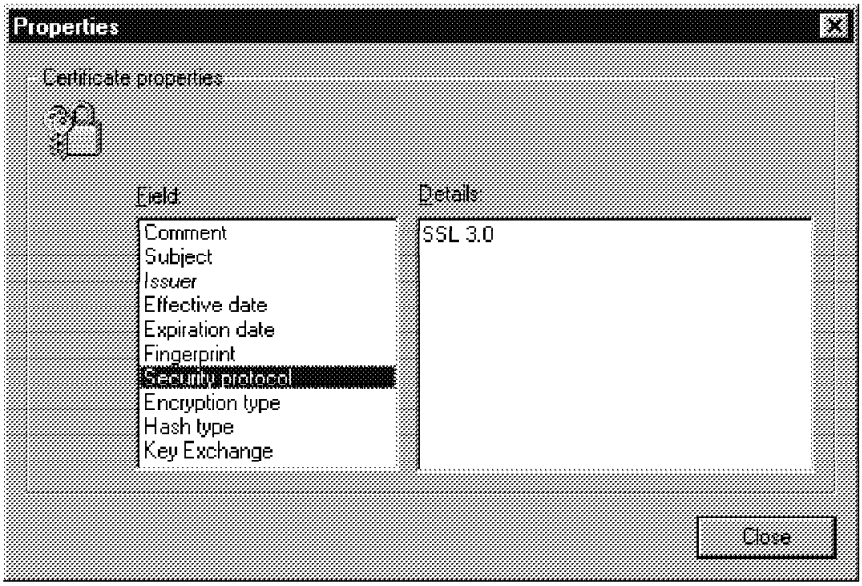


Figure 162. Certificate and Secure Session Information, MS Internet Explorer

9.2 Becoming a Certificate Authority (CA)

Only a CA is able to create a certificate, and while this process is normally handled by a Certificate Authority such as VeriSign and others, you may be required to set up the AS/400 system to become a CA for an internal project, test environment, or simply to enable SSL on the intranet. This section guides you through the steps required to enable an AS/400 system as a CA:

1. You must create a CA public-private key pair and request a CA certificate.
2. The CA certificate must be received into a key ring (CA Key ring).
3. The CA key must be designated as a trusted root.

Before you execute the procedures required to become a CA, study Figure 182 on page 193 to get an understanding of all the parameters and how and where they fit together.

Note

Using this function (acting as a CA) is limited to directly certifying end user-to-end user exchange of data. You are not authorized to issue certificates to third parties or allow others to do so, or to use this function for any other purpose.

9.2.1 Creating a Certificate Request for a CA Certificate

1. From the General Configuration and Administration page, **select the server instance** you are turning into a CA and press the **Change** button. You do not have to set up a specific server instance for the CA but it may be wise to do so.
2. From the Server Instance, select **Configuration and Administration Forms** and scroll down to the Security links.
3. Select **Create Keys**.

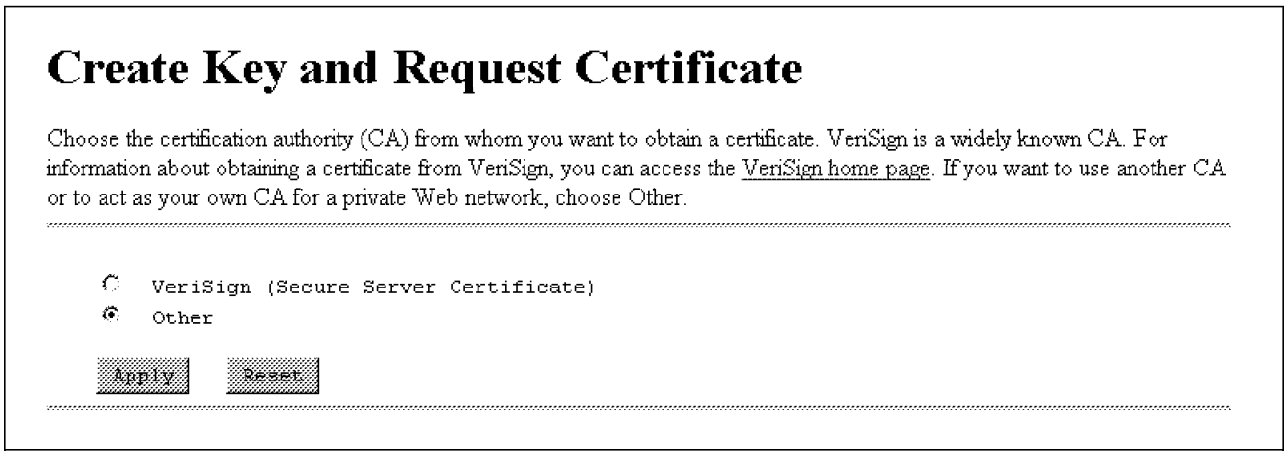


Figure 163. Becoming a CA, Create Key, and Request Certificate

4. Choose **Other** and click **Apply**.

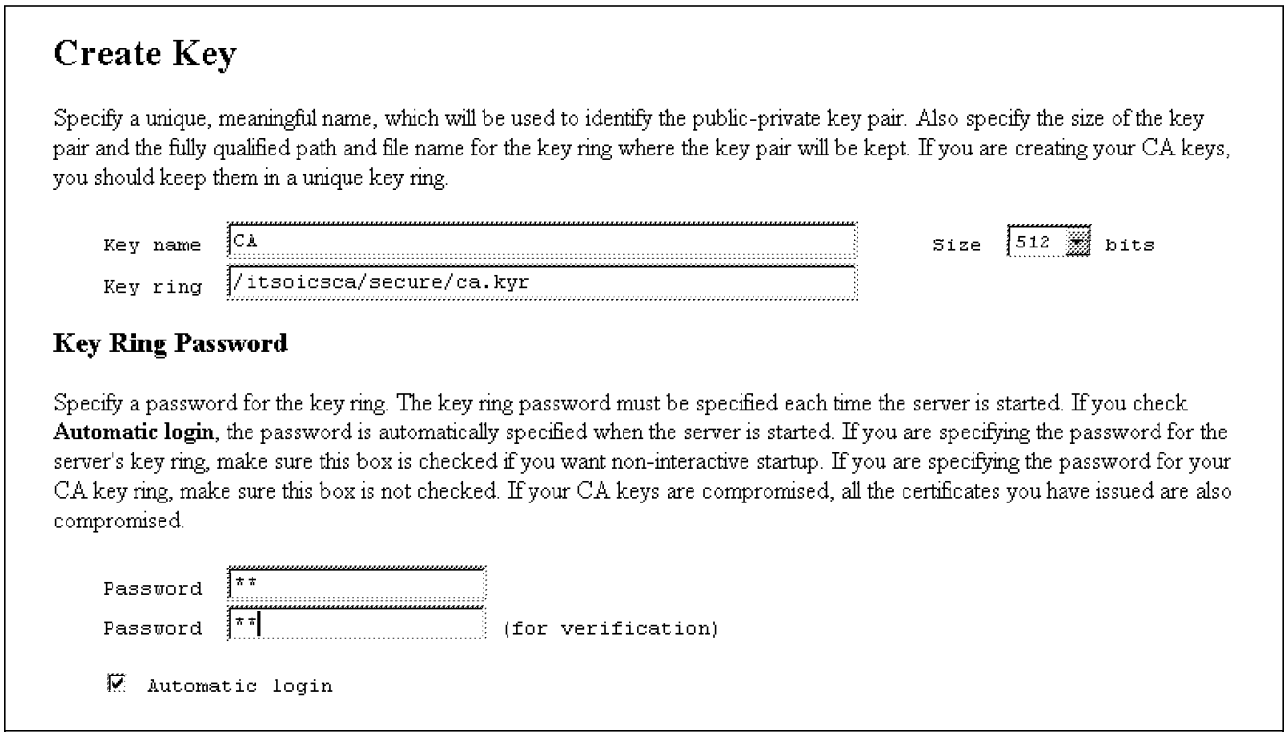


Figure 164. Becoming a CA, Create Key, Key Ring, and Specify Password

5. On the Other Certificate page, enter *your own* CA data:
   - Refer to Section 9.1.2, “Request a Server Certificate” on page 162 for detailed information on the parameters on this page.
   - For **Key name**, specify something meaningful to identify the CA key pair. The key name is also used to label the certificate in the key ring file (*.kyr file).
   - **Key ring** is the fully-qualified path and file name for the CA key ring file. We recommend that you use a separate key ring to keep the CA key pair. The file is created by the server if it does not exist but the directory must exist.
   - The Key ring **password** is case sensitive.

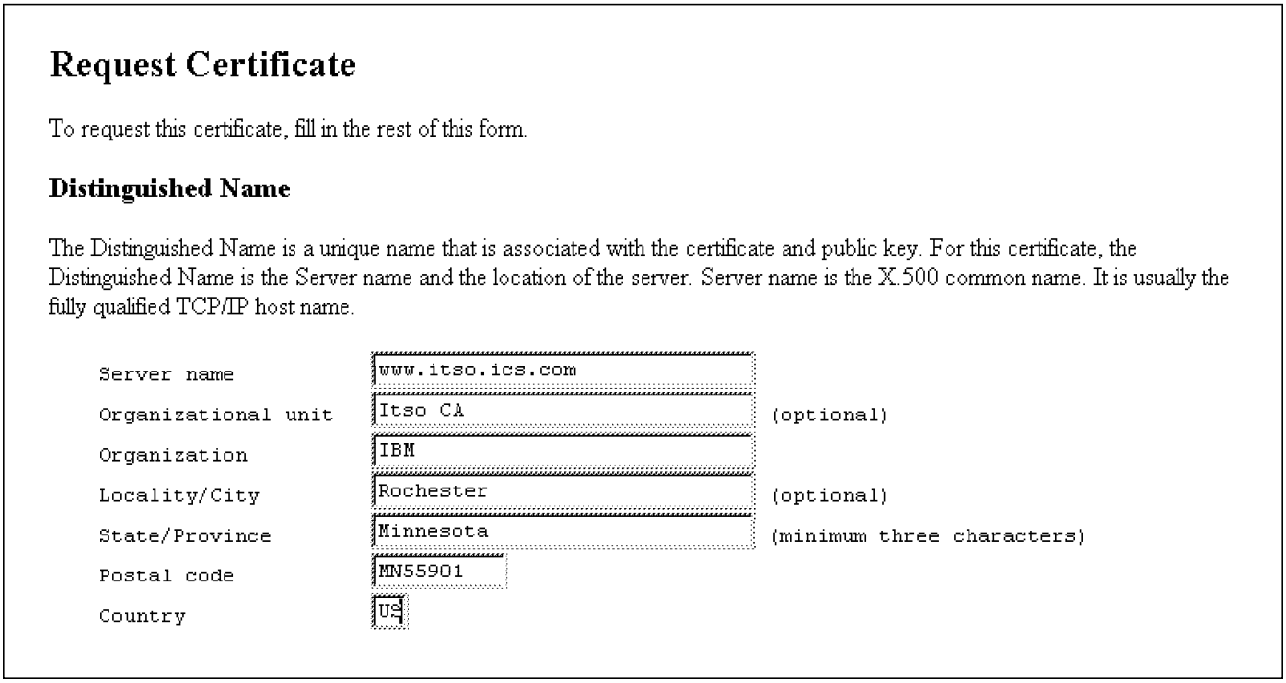


Figure 165. Becoming a CA, CA Distinguished Name

- Fill in the distinguished name information of the CA server. The **server name** is the fully-qualified TCP/IP host name of the server.
- If possible (according to the company policy, and so on), indicate that this **Organizational unit** acts as a CA.
- Enter the **Organization**, the **location**, **State/province**, **Postal or zip code**, and your **Country**.

  See Figure 138 on page 165 for a more detailed explanation of these fields.

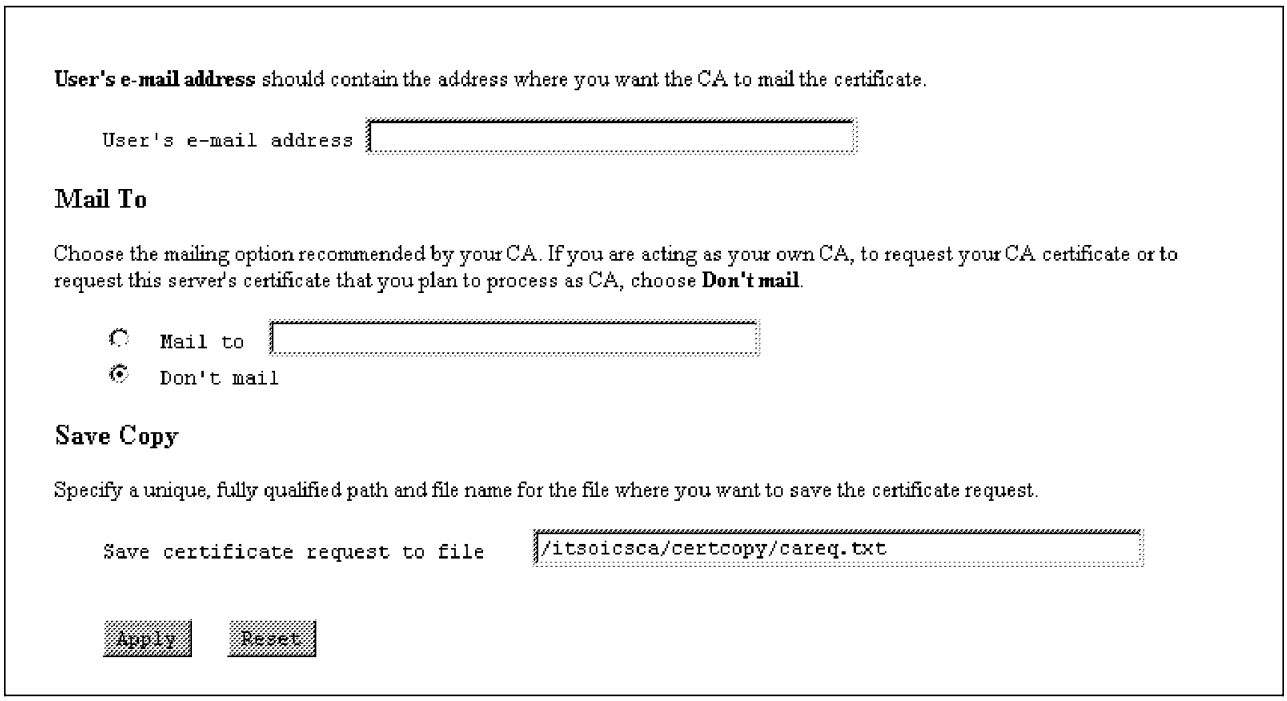


Figure 166. Becoming a CA, E-Mail, and Save Options

- Leave the **User′s e-mail address** open.
- Choose **Don′t mail**.
- Specify the fully-qualified path of a **Save certificate request file**.

  Refer to Section 9.3.3, “A Possible CA /Secure Sub-Directory Structure” on page 196 for a sample directory structure of a CA.
- Read through and verify all the parameters.
- Press the **Apply** button.

6. The last step of the certificate request procedure is the *Confirmation page*.

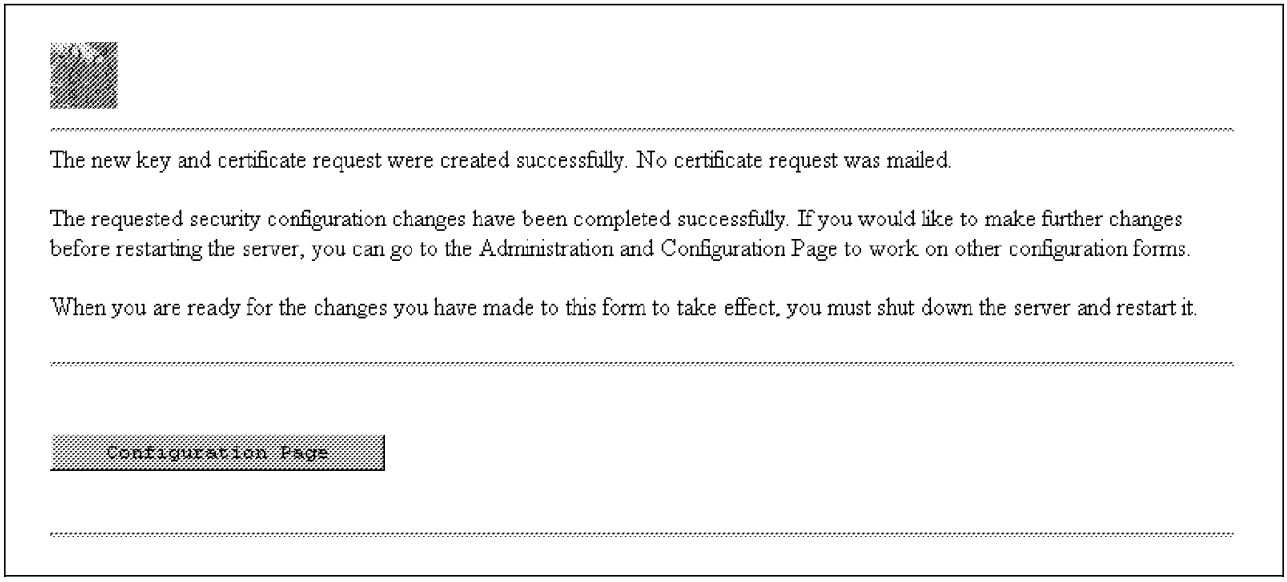


Figure 167. Becoming a CA, New Key, and Certificate Request Confirmation Page

7. Press the **Configuration Page** button to return.

9.2.2 Receiving the CA Certificate

Now that we have created a CA certificate request, we receive that certificate into the key ring file that was created together with the request.

On the Configuration and Administration Form page, select **Receive Certificate** under Security.

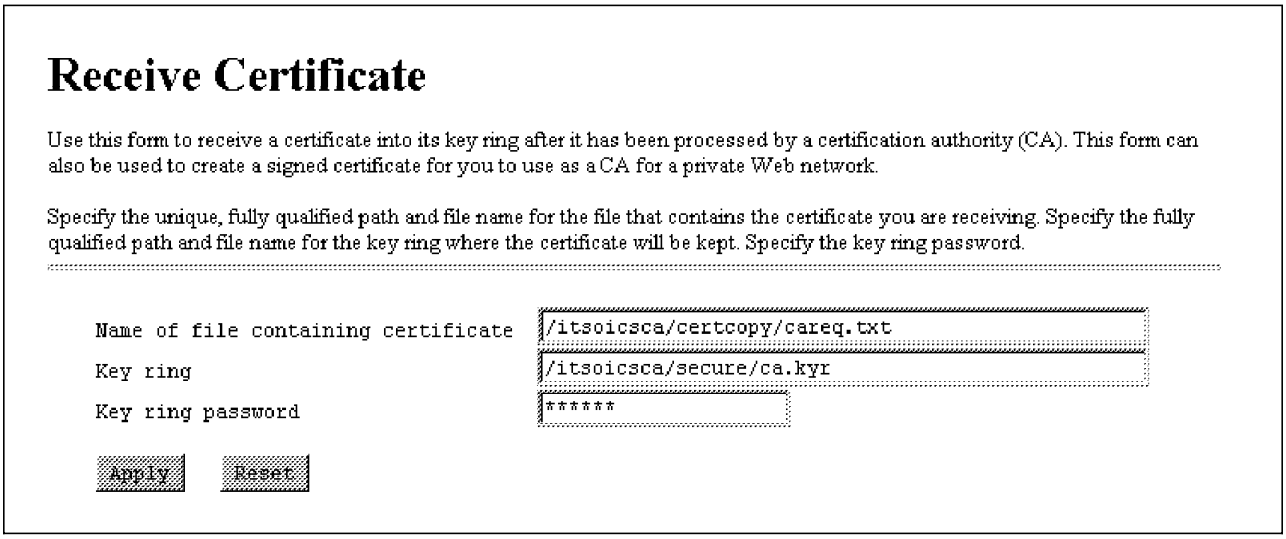


Figure 168. Becoming a CA, Receive Certificate

On the *Receive Certificate* form, enter the following values:

- In the **Name of file containing certificate**, enter the fully-qualified path and name of the certificate request file specified in the *Save certificate request to file* parameter of the *Create key and request certificate* form (Figure 166 on page 184).
- Enter the fully-qualified path and name of the **key ring** file created in the *Create key and request certificate* form (Figure 164 on page 182). This is the CA key ring file.
- The **Key ring password** was also specified on the *Create key and request certificate* form.
- Receive the certificate by pressing the **Apply** button.

On the *Confirmation page*, press the **Configuration Page** button to return.

9.2.2.1 Add Key Ring as Current Key Ring

Before we start processing certificates, we must designate the CA key ring as the current key ring and open up the server instance for SSL connections.

- From the Configuration and Administration Form page, select **Security Configuration** under Security.

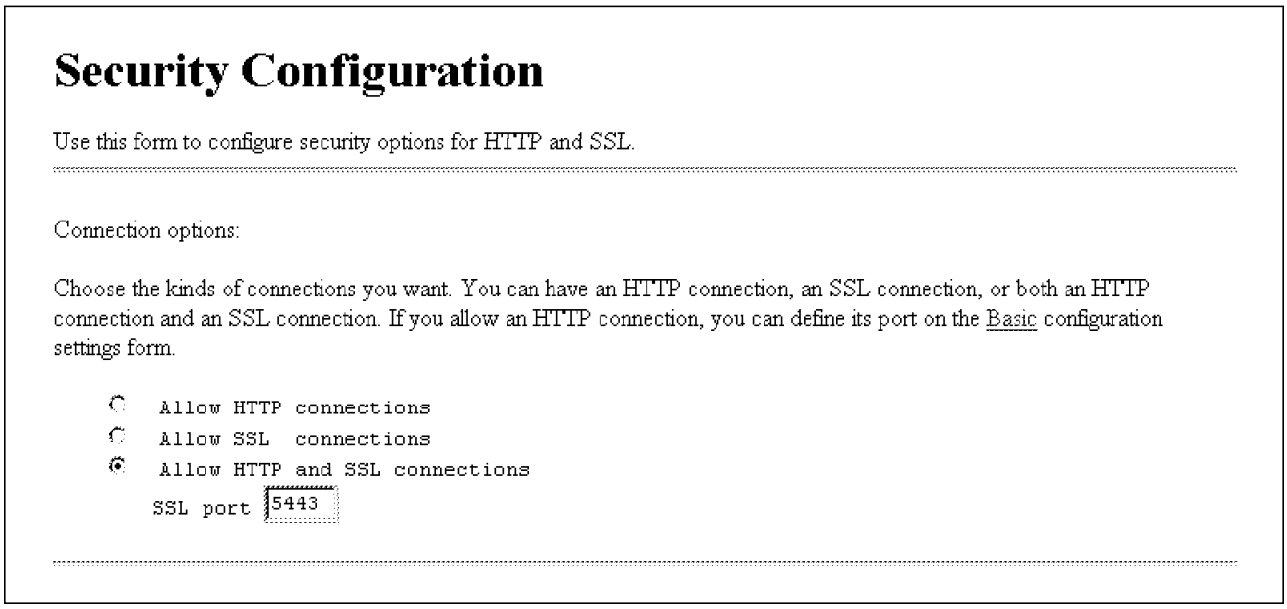


Figure 169. Becoming a CA, Allow SSL, and Specify SSL Port

- Choose **Allow HTTP and SSL connections**.
- Specify a port for the SSL connections. Specify a port number higher than 1024.
- Click on the **Add key ring** button.

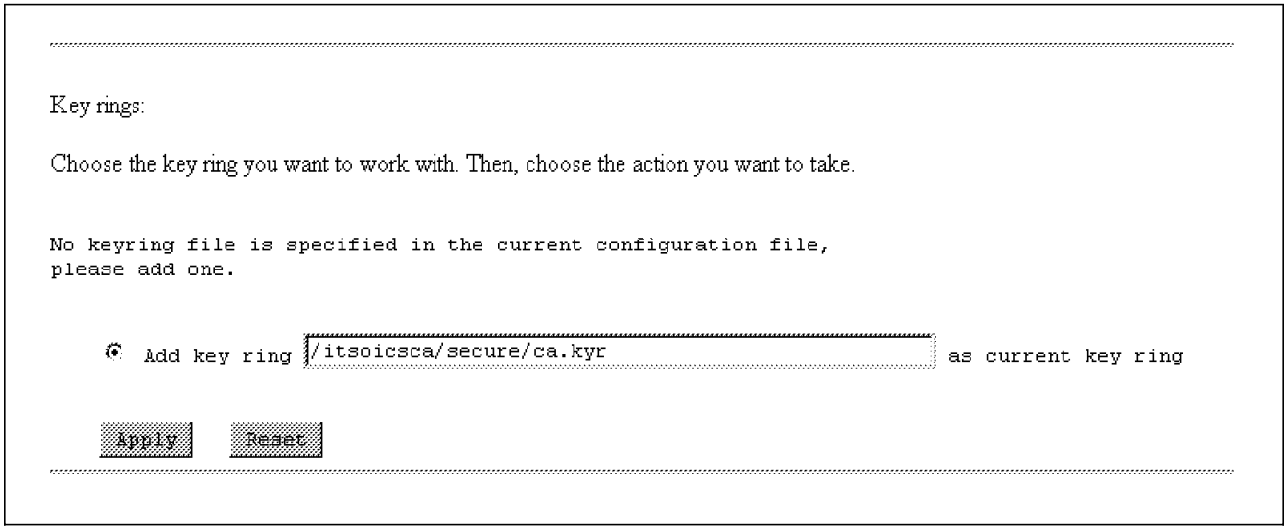


Figure 170. Becoming a CA, Add CA's Key Ring as Current Key Ring

- Enter the fully-qualified path and name of the CA key ring file (see Figure 164 on page 182) to add it **as current key ring**.
- Press the **Apply** button and wait for the *Confirmation page*.

Figure 171. Allow HTTP and SSL Confirmation Page

- Note that the four ICSS for AS/400 directives have been set.
- Press the **Configuration Page** button to return.

9.2.3 Designate as a Trusted Root

For the AS/400 system to create a certificate, it must be designated as a trusted root. In this step, we self-sign the certificate we previously received and designate it as a trusted root.

1. From the Configuration and Administration Form page, select **Key Management** under Security.

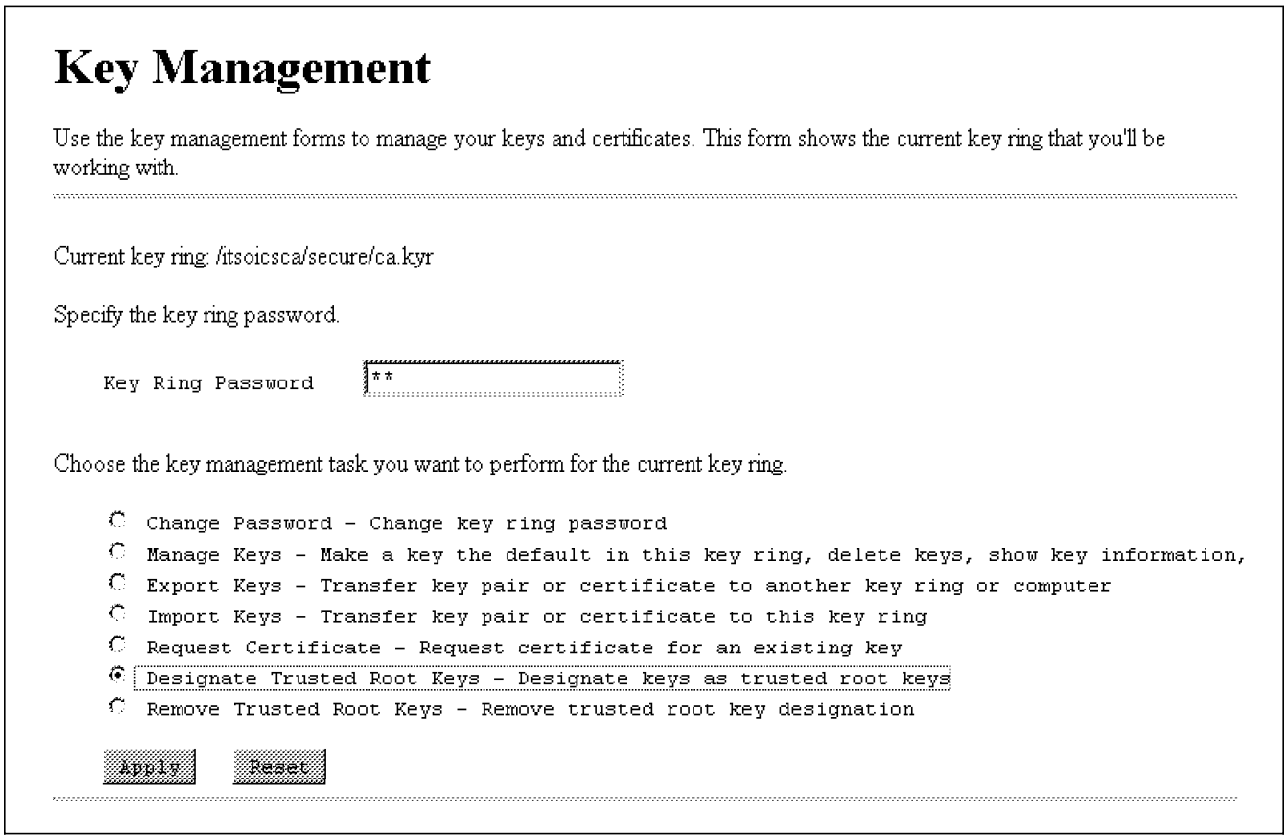


Figure 172. Becoming a CA, Designate a Trusted Root

2. On the Key Management page, enter the CA **Key Ring Password** and remember that this password is *case sensitive*. Note that the current key ring is the CA key ring.
3. Choose the **Designate Trusted Root Keys** option and press the **Apply** button. The following page is shown (Figure 173 on page 189).

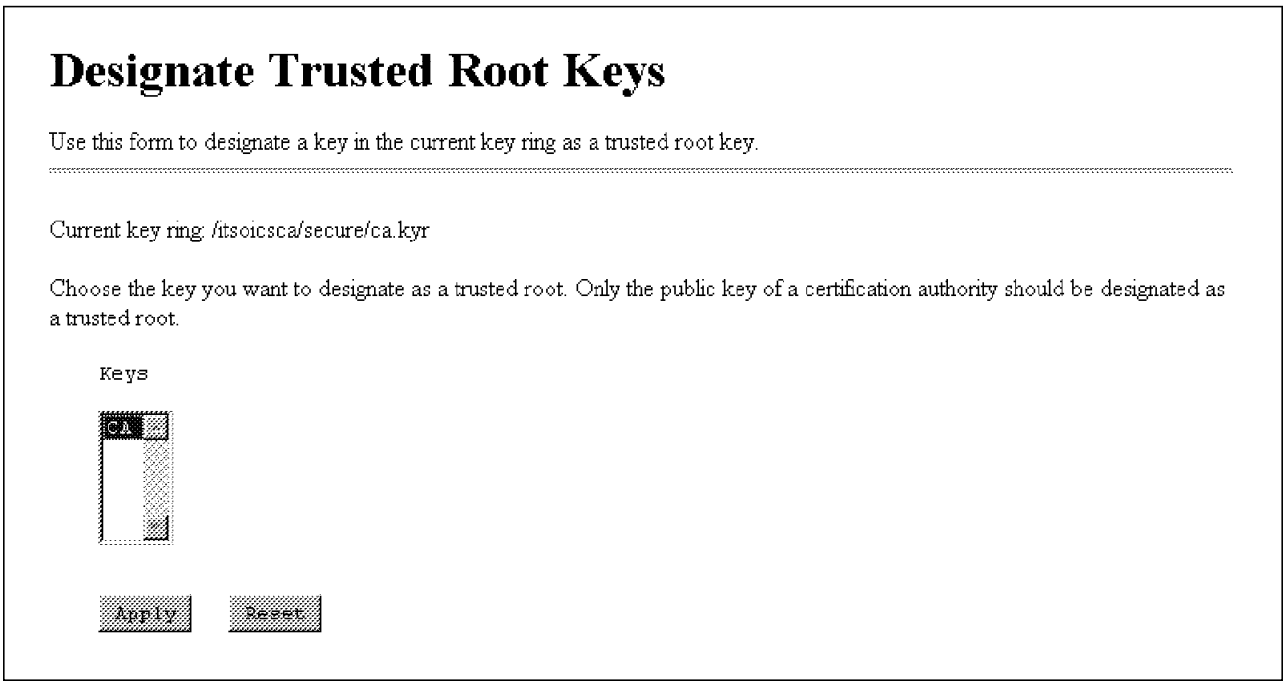


Figure 173. Becoming a CA, Designate CA Key as Trusted Root

4. Select the **CA key** in the **Keys** box and press **Apply**.

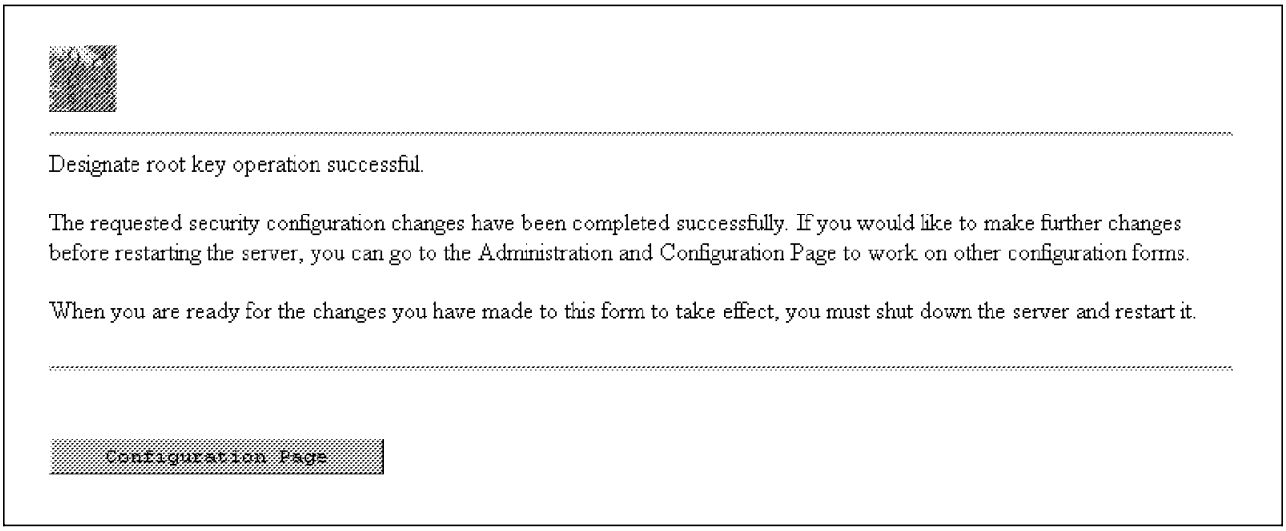


Figure 174. Becoming a CA, Confirm Trusted Root Designation

5. The *Confirmation page* indicates a successful Designate trusted root operation. Press the **Configuration Page** button to return.

The CA key has now been designated as a trusted root.

9.2.4 Verifying the CA Setup

At this stage, we are a CA and may process certificates for this or other AS/400 systems. Go through the following procedure to verify the CA capabilities.

1. From the Configuration and Administration Form page, select **Key Management** under Security.

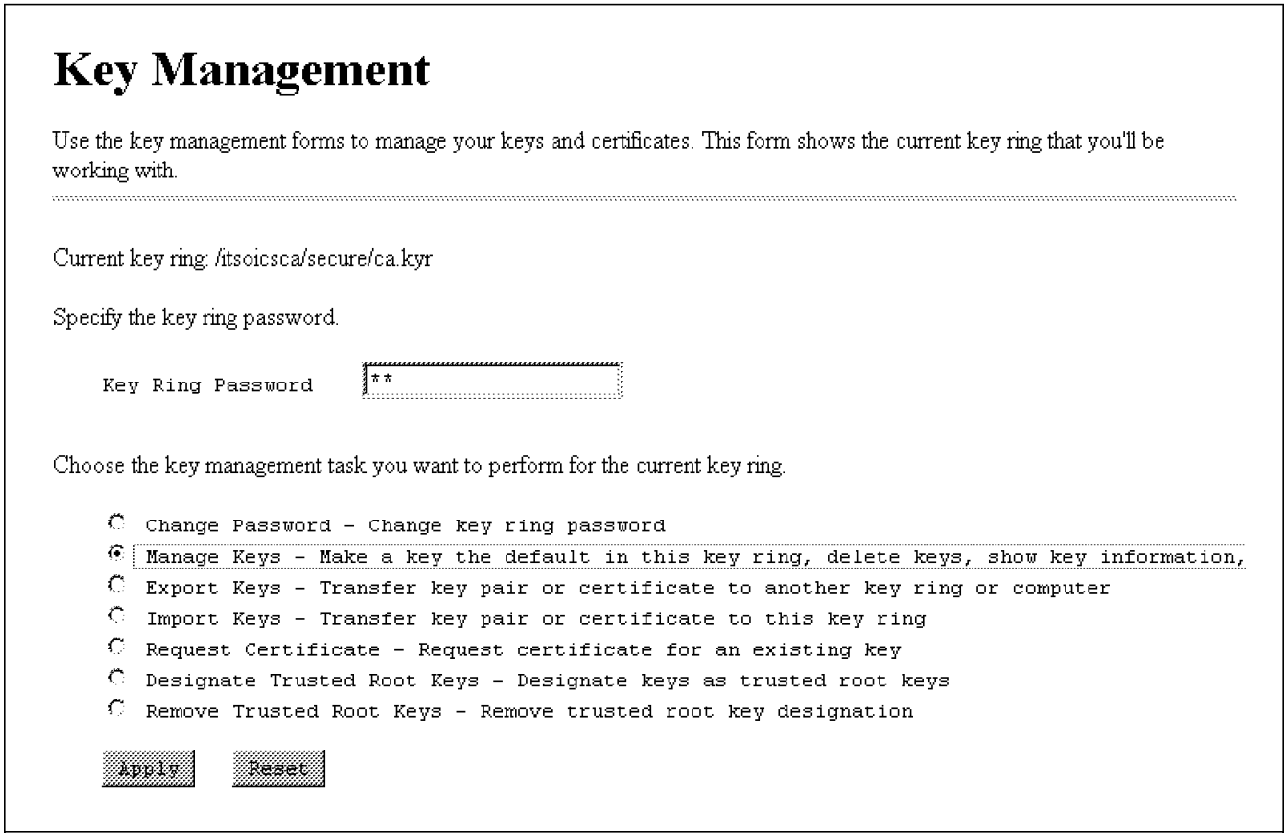


Figure 175. Key Management, Select Manage Keys

2. On the Key Management page, enter the CA **Key Ring Password** (remember that this password is still *case sensitive*).
3. Choose **Manage Keys** option and press the **Apply** button. The following page is shown (Figure 176 on page 191).

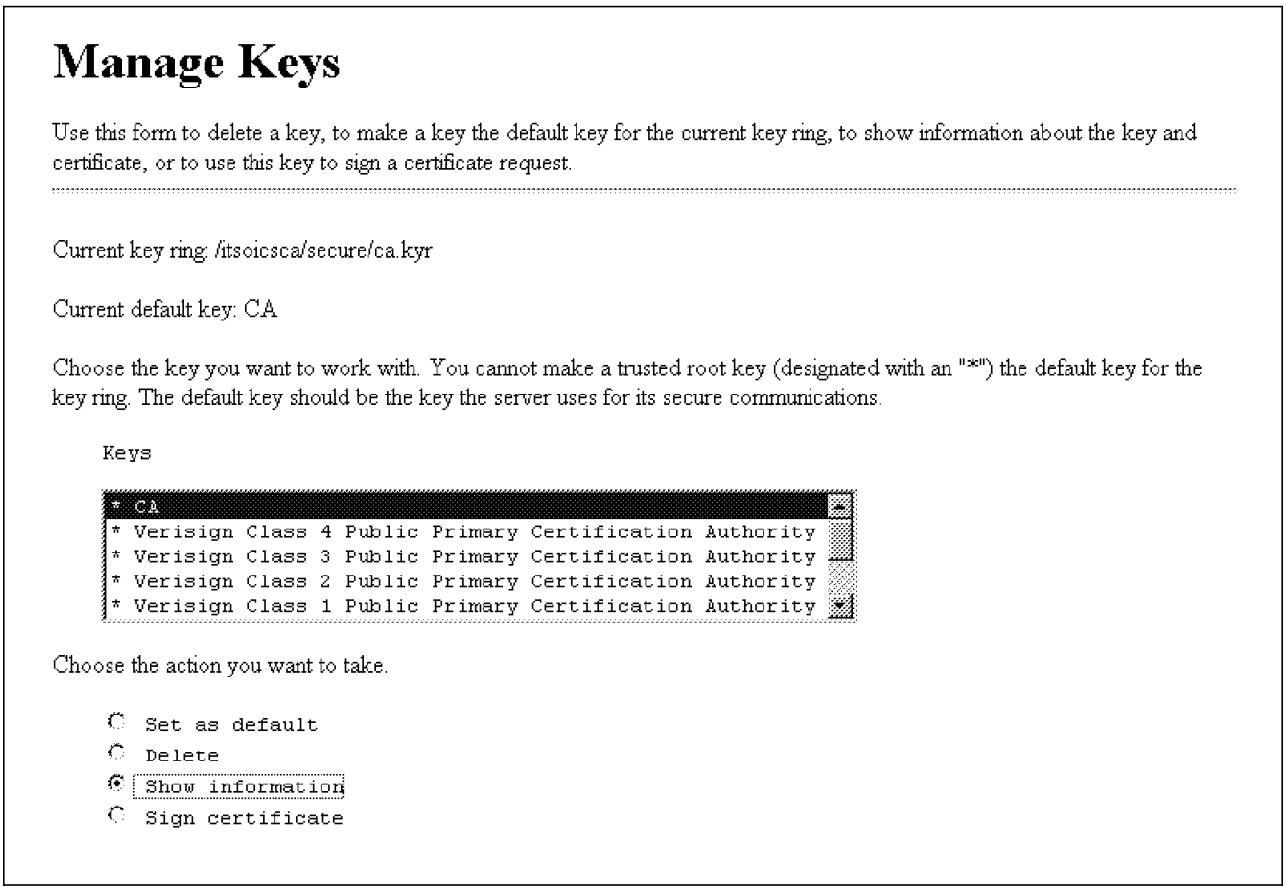


Figure 176. Manage Keys, Select Show Information

4. Select the CA key from the **Keys** box, choose **Show information**, and press **Apply**.

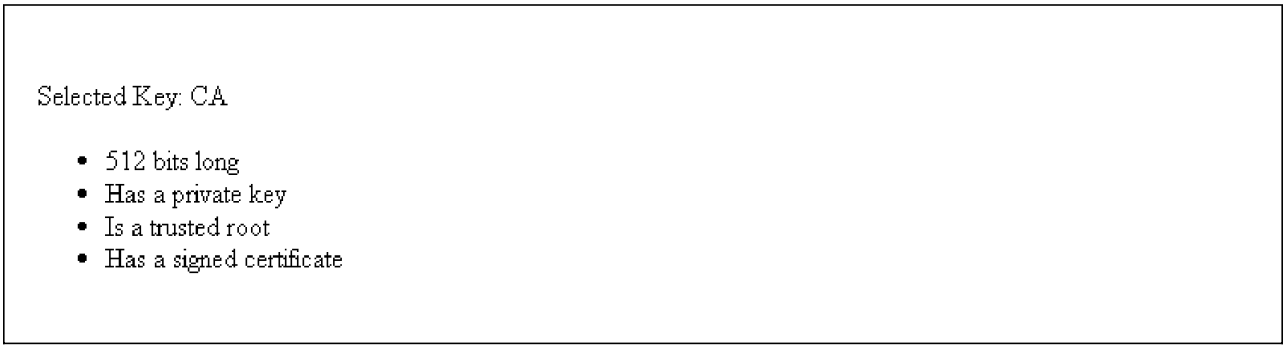


Figure 177. CA Key Part One

- The key length depends upon the licensed program (LP) you have installed, 5769-NC1 (U.S. and Canada) or 5769-NCE (International).
- Note that the CA key is *a trusted root* and that it *has a signed certificate*.

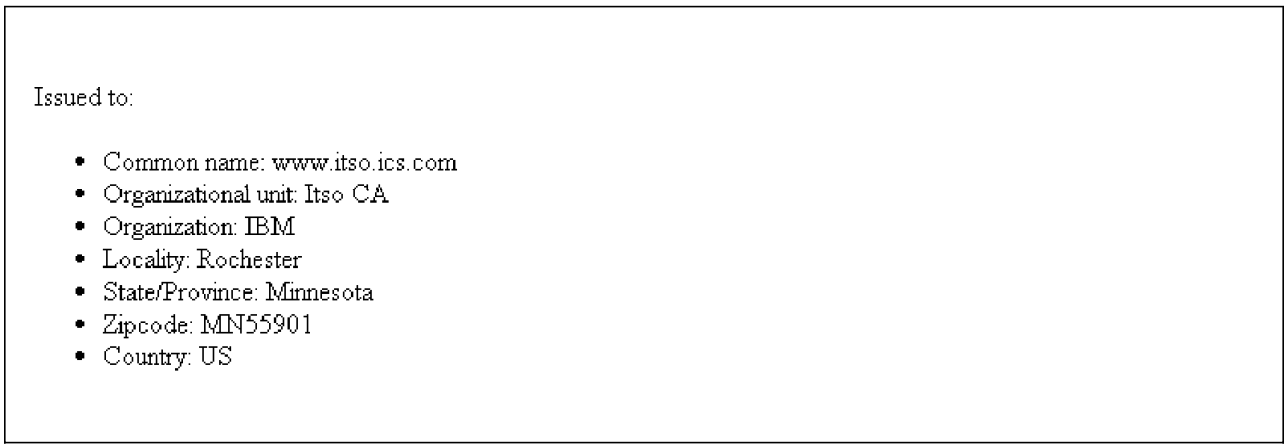


Figure 178. CA Key Part Two

- As the key has a self-signed certificate, it has been *Issued to* (preceding figure) and *Issued by* (following figure) the same *Organizational unit* (in this case, *Itso CA*).

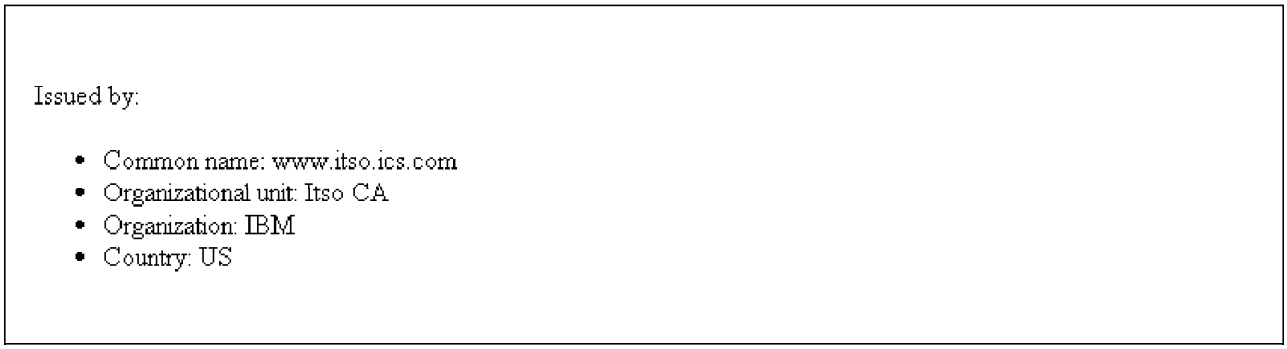


Figure 179. CA Key Part Three

- The certificate information contains the certificate serial number and expiration date. Remember to renew the certificate before it expires.

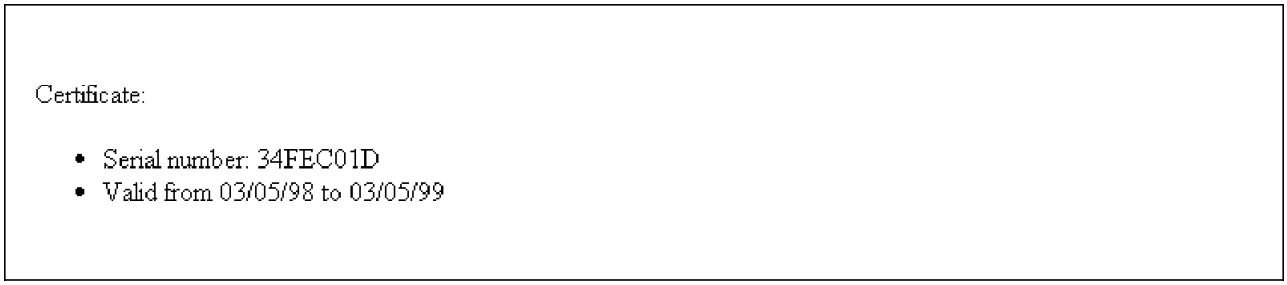


Figure 180. CA Key Part Four

Press the **I** "Information icon" to view the online help of the Manage Keys form.

5. Return to the Configuration Page by pressing the **return** icon.

9.2.4.1 New Directives in the HTTP Config File

If we look at the CA HTTP configuration file, WRKHTTPCFG CFG(CA), we see these directives.

```
              HTTP Configuration File

         normalmode       on
         sslmode       on
         sslport       5443
         keyfile /itsoisca/secure/ca.kyr
```

Figure 181. SSL Directives Added to HTTP Configuration File

The following figure (Figure 182) gives you a better understanding of how the parameters interconnect when you set up the server to act as a CA.

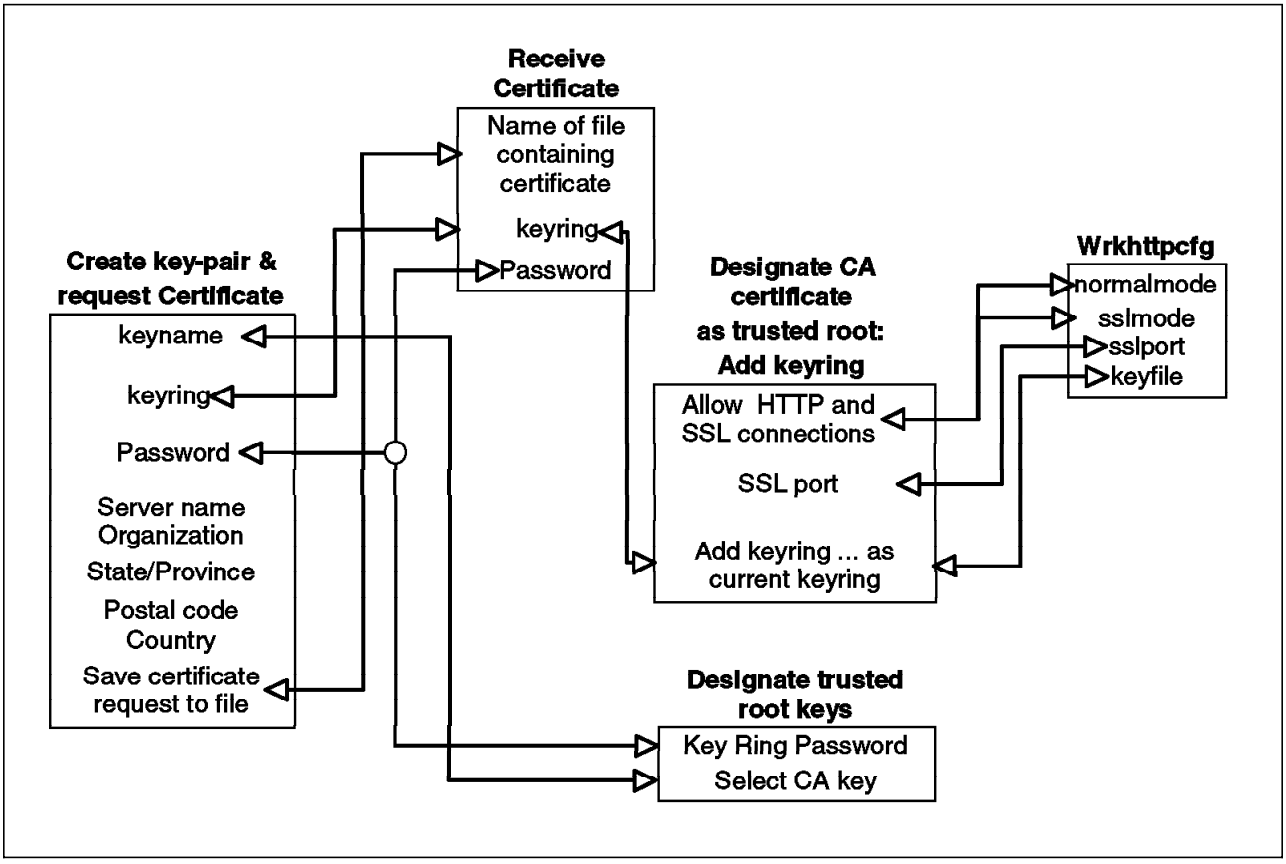


Figure 182. Becoming a CA, How the Different Parameters Interconnect

9.3 Acting as a Certificate Authority

When the server has been set up as a CA, you may sign certificates for other servers in an intranet. This is normally a two-step procedure:

- First you sign the *Certificate request file* received from the server.
- Then you return the signed server certificate together with the self-signed CA certificate to the server requesting the certificate.

On the requesting server, first receive the self-signed CA certificate then the server certificate into the server key ring file.

9.3.1 Sign a Certificate Request

From the Configuration and Administration Forms page, select **Key Management** under security.

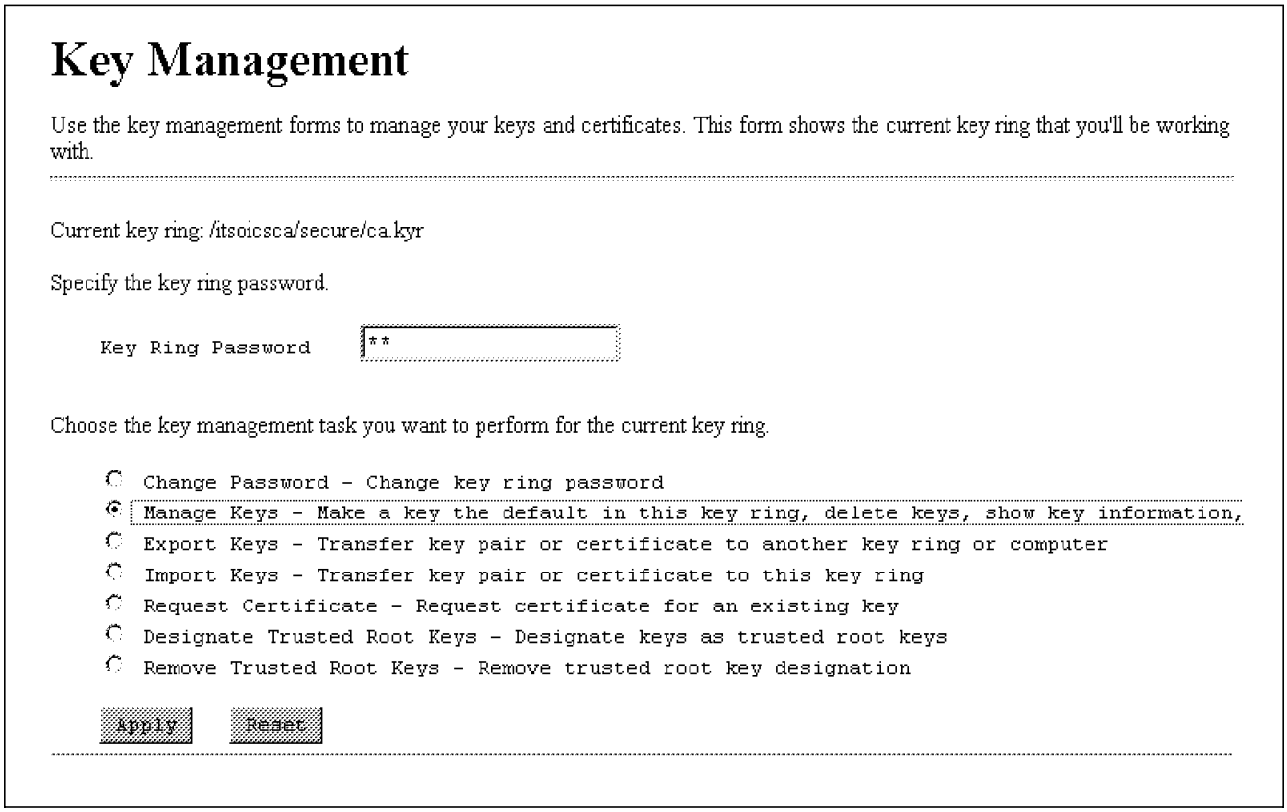


Figure 183. Sign Certificate, Step One

From the Key Management page, type the **Key Ring Password**, select **Manage Keys**, and then press **Apply**.

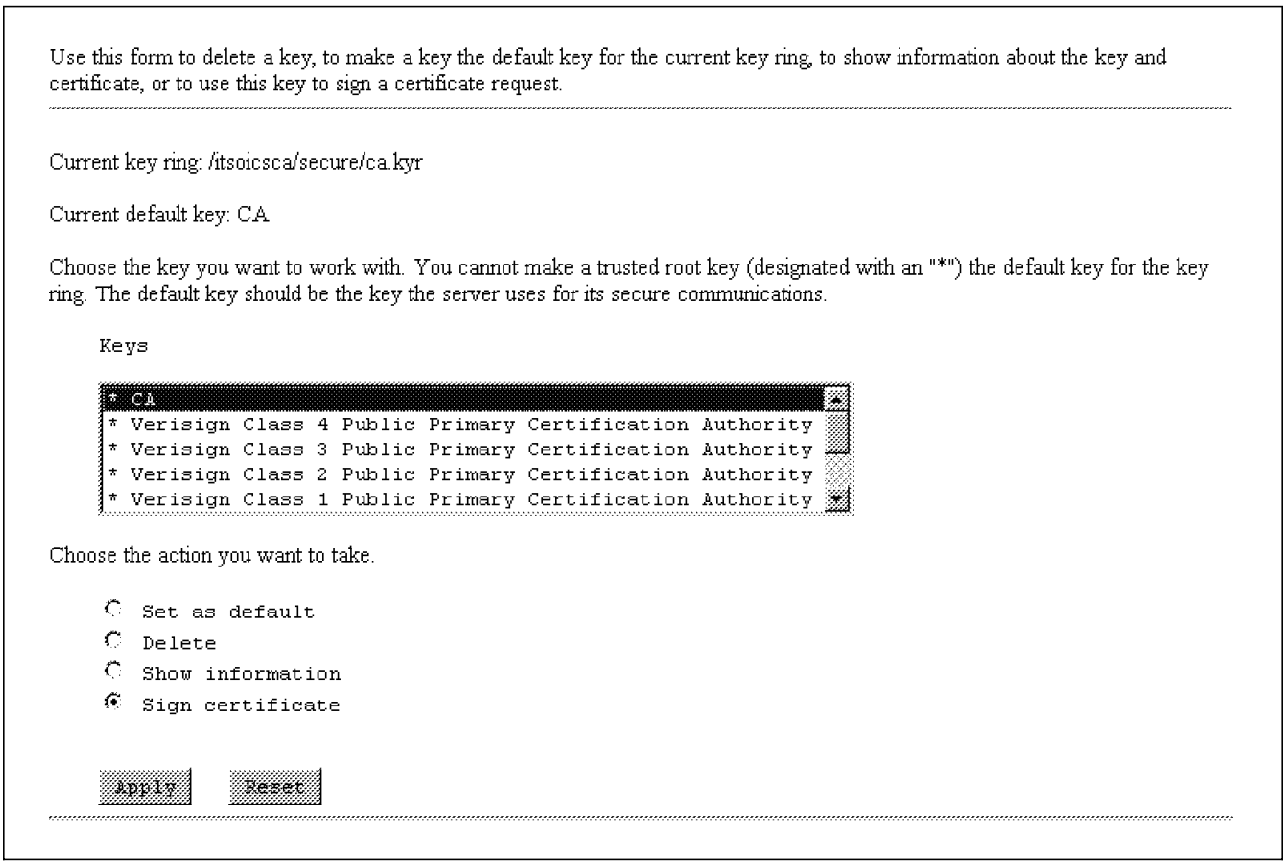


Figure 184. Sign Certificate, Step Two

From the Manage Keys page, select the **CA** key, choose **Sign certificate**, and press **Apply**.

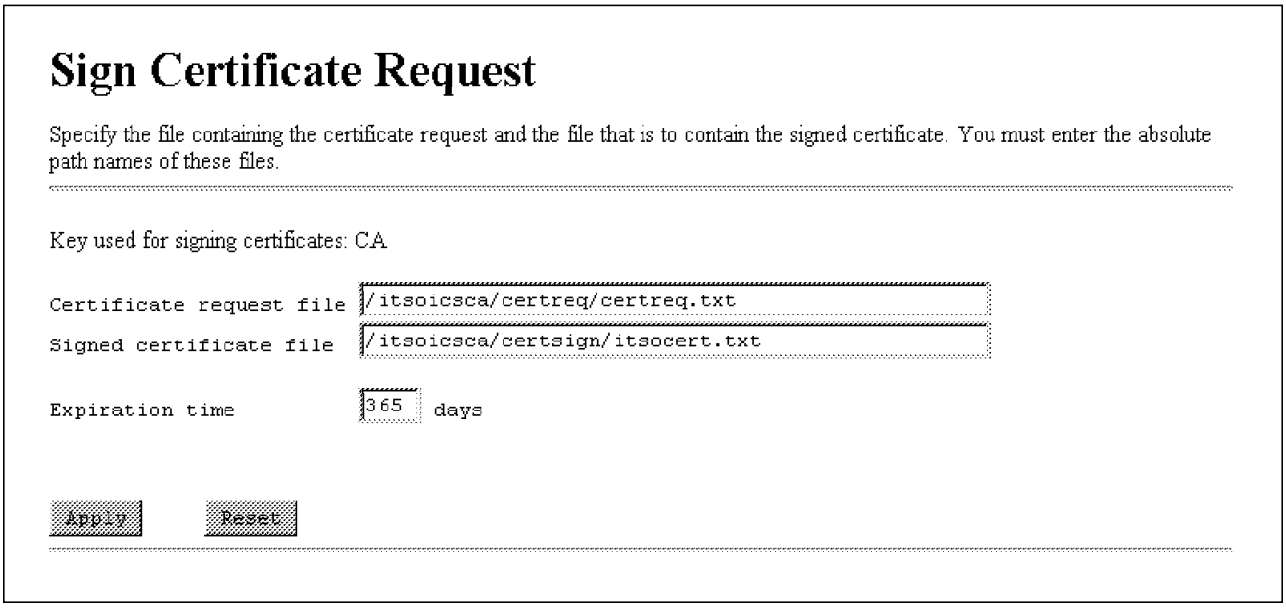


Figure 185. Sign Certificate, Step Three

On the Sign Certificate Request page, enter the fully-qualified path and file name for the **Certificate request file** and the **Signed certificate file**.

The *Certificate request file* is the file that is received from the server requesting a signed certificate.

The *Signed certificate file* is the file where you store the signed certificate.

Specify a number of days in the **Expiration time** box and press **Apply**.

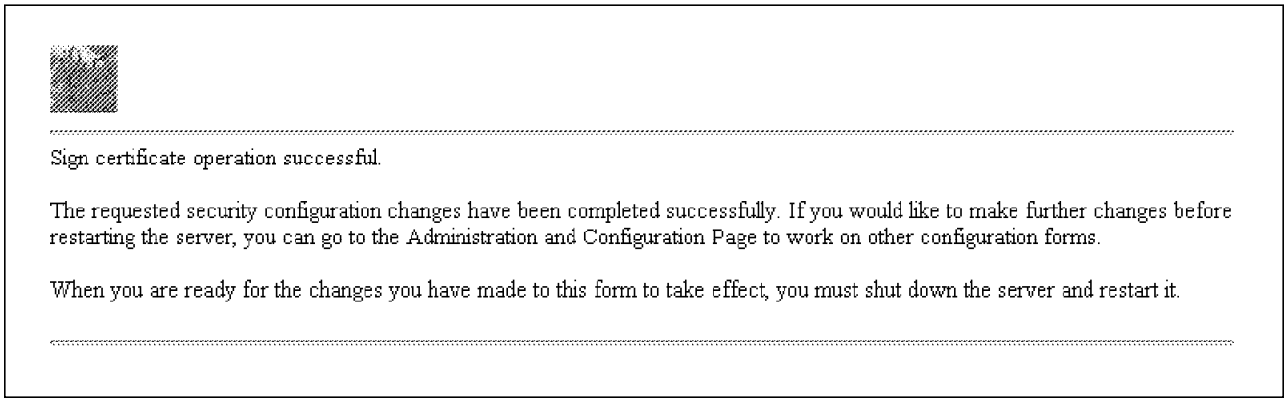


Figure 186. Sign Certificate, Step Four

The *Sign certificate operation successful* page is displayed. Press the **Configuration Page** button to return.

9.3.2 Send the Self-Signed CA Certificate and Server Certificate to the Server

Before a server can use a certificate, the server must have the CA that issued the certificate as a trusted root. In addition to placing the server certificate in a file, we, therefore, also have to put the self-signed CA certificate in a file.

The saved CA certificate request file (*Save certificate request to file* in Figure 166 on page 184) and the server certificate file (*Signed certificate file* in Figure 185 on page 195) must now be sent to the server. Refer to Figure 143 on page 169 through Figure 147 on page 172 for an example of the procedure required to receive the certificates.

Note: These files should be sent to the server by two different routes to prevent an adversary intercepting them. The files are digitally signed and so cannot be modified unoticed but an adversary could use the files themselves or exchange them.

9.3.3 A Possible CA /Secure Sub-Directory Structure

If you are acting as a CA for many servers, you can benefit from setting up a good CA directory structure. You need two or three subdirectories in the CA secure directory.

CertReq The CertReq directory stores all *certificate requests* that you receive from the other servers in your intranet.

SignCert The SignCert directory is used for all the certificates that you *sign as a CA*.

If you find it necessary, you may also want a separate directory for the CA certificate to be sent to servers in your intranet.

CACert This directory stores the *CA certificate* request file.

Keep in mind that these directories must have a *high level of protection*.

9.4 How To′s

In this section, there is a series of ″quick lists″ to achieve:

- Enabling a Secure Connection on an Intranet AS/400.
- Requesting and Receiving a Certificate from a Third Party CA.
- Acting as a CA: Process a server certificate request.

9.4.1 Enabling a Secure Connection on an Intranet AS/400

Follow these steps to enable the AS/400 system to act as a secure server in an intranet *without* an intranet CA. The AS/400 system acts as its own CA. The procedure is divided into two parts:

- How to become your own CA.
- Use your new CA ability to create a certificate.

9.4.1.1 Becoming Your Own CA - Step 1

1. From the ADMIN server General Configuration and Administration page, select your server instance and press the **Change** button.
2. On the Configuration and Administration Form page, select **Create Keys** under Security.
3. On the Create Key and Request Certificate page, select **Other** and press the **Apply** button.
4. On the Other Certificate page, enter the following values:
   - Enter the CA Key Name (*mycakey*).
   - Type the fully-qualified **Key ring** file name (*/myserver/secure/mycakey.kyr*).
   - Type the password for the key ring file in both password fields.
   - Leave the Automatic login box checked.
5. Fill in the following fields for **Distinguished Name**:
   - Server name
   - Organization unit or department
   - Organizational name
   - Locality/City
   - State/Province
   - Postal code
   - Country
   - Leave **User′s e-mail address** blank.
   - Select **Don′t mail**.
   - Save the request as: */myserver/secure/mycareq.txt*.
6. Press the **Apply** button.

Wait for the confirmation page and press the **Configuration Page** button.

9.4.1.2 Becoming Your Own CA - Step 2

1. On the Configuration and Administration Form page, select **Receive Certificates**.
2. Enter the following on the Receive Certificate form:
   - Enter the file name for the certificate request (*/myserver/secure/mycareq.txt*) in the field **Name of file containing certificate**.
   - Enter the key ring file (*/myserver/secure/mycakey.kyr*) as the Key Ring file name.
   - Type the “mycakey” password in the **Key ring file password** field.
3. Press the **Apply** button.

Wait for the confirmation page and press the **Configuration Page** button.

9.4.1.3 Becoming Your Own CA - Step 3

1. On the Configuration and Administration Form page, select **Security Configuration**.
2. On the Security Configuration page, select **Allow HTTP and SSL connections**. Then enter the port number for SSL connections.
3. Select **Add key ring** and type the key ring file (*/myserver/secure/mycakey.kyr*) as the **current key ring** file name.
4. Press the **Apply** button.
5. Wait for the confirmation page and press the **Configuration Page** button.
6. On the Configuration and Administration Form page, select **Key Management**.
7. On the Key Management page, type the **Key ring password**.
8. Choose the **Designate Trusted Root Keys** option and press the **Apply** button.
9. On the Designate Trusted Root Keys page, verify that the CA key (*mycakey*) is selected and press the **Apply** button.
10. Wait for the confirmation page and press the **Configuration Page** button.

The server is now a CA. Follow these steps to view the CA certificate:

- On the Configuration and Administration Form, select **Key Management**.
- On the Key Management form, enter the CA **Key ring password**, select **Manage Keys**, and press the **Apply** button.
- On the Manage Keys form, verify that the CA key is selected, select **Show information**, and press the **Apply** button.

9.4.1.4 Request and Receive a Certificate - Step 1

1. On the Configuration and Administration Form page, select **Create Keys** under Security.
2. On the Create Key and Request Certificate page, select **Other** and press the **Apply** button.
3. On the Other Certificate page, enter the following values:
   - The secure server Key Name (*mysecurekey*).

- Type the fully-qualified **Key ring** file name (*/myserver/secure/myserverkey.kyr*).
- Type the password for the key ring file in both password fields.
- Leave the Automatic login box checked.

4. Fill in the following fields for **Distinguished Name**:
   - Server name
   - Organization unit or department
   - Organizational name
   - Locality/City
   - State/Province
   - Postal code
   - Country
   - Leave **User′s e-mail address** blank.
   - Select **Don′t mail**.
   - Save the request as: */myserver/secure/myserverreq.txt*.
5. Press the **Apply** button.
6. Wait for the confirmation page and press the **Configuration Page** button.

9.4.1.5 Request and Receive a Certificate - Step 2

1. On the Configuration and Administration Form page, select **Key Management**.
2. On the Key Management page, enter the following values:
   - Type the CA key ring password, select **Manage Keys**, and press the **Apply** button.
3. On the Manage keys form, verify that the CA key is selected, choose **Sign certificate**, and press the **Apply** button.
4. On the Sign Certificate Request form, enter the following values:
   - Type the file name of the secure server certificate request file (*/myserver/secure/myserverreq.txt*) as the **Certificate request file**.
   - Type the file name of the new certificate in the **Signed certificate file** field: */myserver/secure/mycertificate.txt*.
   - Leave 365 days in the **Expiration date** field.
5. Press the **Apply** button.
6. Wait for the confirmation page and press the **Configuration Page** button.

9.4.1.6 Request and Receive a Certificate - Step 3

1. On the Configuration and Administration Form page, select **Security Configuration**.
2. Select **Add key ring** and type the **server key ring** file (*/myserver/secure/myserverkey.kyr*) as the **current key ring** file name.
3. Press the **Apply** button.
4. Wait for the confirmation page and press the **Configuration Page** button.

9.4.1.7 Request and Receive a Certificate - Step 4

1. On the Configuration and Administration Form page, select **Receive Certificate**.
2. On the Receive Certificate page, enter the following values:
   - Type the name of the CA certificate request file (*/myserver/secure/mycareq.txt*) in the **Name of file containing certificate**.
   - Type the name of the server key ring file (*/myserver/secure/myserverkey.kyr*) in the field **Key ring**.
   - Type the secure server key ring password as the **password** for this ring.
3. Press the **Apply** button.
4. Wait for the confirmation page and press the **Configuration Page** button.

9.4.1.8 Request and Receive a Certificate - Step 5

1. On the Configuration and Administration Form page, select **Key Management**.
2. On the Key Management page, type the secure server key ring password, select **Designate Trusted Root Keys**, and press the **Apply** button.
3. On the Designate Trusted Root Keys form, select the CA trusted root from the listbox and press the **Apply** button. Wait for the confirmation page and press the **Configuration Page** button.

9.4.1.9 Request and Receive a Certificate - Step 6

1. On the Configuration and Administration Form page, select **Receive Certificate**.
2. On the Receive Certificate page, enter the following values:
   - Type the name of the server certificate file (*/myserver/secure/mycertificate.txt*) in the **Name of file containing certificate** field.
   - Type the name of your secure server key ring (*/myserver/secure/myserverkey.kyr*) in the field **Key ring**.
   - Type the secure server key ring **password**.
3. Press the **Apply** button.
4. Wait for the confirmation page and press the **Configuration Page** button.

The server can now establish a secure session using the self-signed certificate.

9.4.2 Requesting and Receiving a Certificate from a Third Party CA

Follow these steps to request and receive a certificate from a third-party CA (for example, from Verisign).

9.4.2.1 Creating a Certificate Request

1. On the Configuration and Administration Form page, select **Create Keys** under Security.
2. On the Create Key and Request Certificate page, select **VeriSign** or **Other** and press the **Apply** button.
3. On the Verisign or Other Certificate page, enter the following values:
   - The secure server Key Name (*mysecurekey*).
   - Type the fully-qualified **Key ring** file name (*/myserver/secure/myserverkey.kyr*).
   - Type the password for the key ring file in both password fields.
   - Leave the Automatic login box checked.
4. Fill in the following fields for **Distinguished Name**:
   - Server name
   - Organization unit or department
   - Organizational name
   - Locality/City
   - State/Province
   - Postal code
   - Country
   - Enter your e-mail address if your CA requests an e-mail address.
   - Select (and fill in) the appropriate **Mail to** parameters.
   - Save the request as: */myserver/secure/myserverreq.txt*.
5. Press the **Apply** button.
6. Wait for the confirmation page and press the **Configuration Page** button.

9.4.2.2 Receiving a Signed Certificate

1. On the Configuration and Administration Form page, select **Receive Certificate** under Security.
2. On the Receive Certificate form, enter the following values:
   - The fully-qualifed path and name of the file with the signed certificate (*/myserver/secure/mysignedcertificate.txt*). Where ″/myserver/secure/mysignedcertificate.txt″ is the name and location (directory) of the signed server certificate file received from the CA.
   - The fully-qualifed path and name of the key ring file you are receiving the certificate into (*/myserver/secure/myserverkey.kyr*).
   - Type the key ring file password.
3. Press the **Apply** button.
4. Wait for the confirmation page and press the **Configuration Page** button.

The server can now establish a secure session using the third-party certificate.

9.4.3 Acting as a CA: Process a Server Certificate Request

Follow these steps to sign a certificate as a CA.

1. From the Configuration and Administration Forms page, select **Key Management**.
2. Enter the key ring password, select **Manage Keys**, and press **Apply**.
3. On the Manage Keys form, select the CA key and click **Sign certificate**; then press **Apply**.
4. On the Sign Certificate Request form:
   - Type the fully-qualified path and name of the certificate request file.
   - Type the fully-qualified path and name of the file where you want the signed certificate to go.
5. Click **Apply**, wait for the **Confirmation Page**, and press the **Configuration Page** button.
6. Send the *Signed certificate file* and the *saved CA certificate request file* (*/myserver/secure/mycareq.txt*) to the certificate requester.

9.4.4 Secure Server Planning Form

The following forms may be helpful when planning a secure server configuration.

Table 6. Secure Server Planning Form	
Server instance	
Secure server or CA	

Parameters	Your Values	Secure Server	CA
Key name		Y	Y
Key ring file		Y	Y
Key ring password	******	Y	Y
Server name		Y	Y
Organizational unit		Optional	Optional
Organization		Y	Y
Locality/City		Optional	Optional
State/province		Y	Y
Postal code		Y	Y
Country		Y	Y
User′s e-mail address		Y	N
Mail to e-mail address		Y	N
Save certificate request as:		Y	Y
Enable SSL		Y	Optional
Enable HTTP		Optional	Optional
SSL port number		Y	Optional

Table 7. Receiving Signed Certificate	
Parameter	**Your value**
Received certificate file	

Table 8. Acting as a CA	
Parameters	**Your Values**
Received certificate request file name	
Signed certificate file name	
CA certificate request file name	

Chapter 10. Emulator Products

In this chapter, we are going to look at 5250 emulation over the Internet.

A Web browser uses HTML to display information to the user. To display 5250 screens on a Web browser, some kind of translation facility is, therefore, required to convert the 5250 to HTML. An HTML gateway provides this facility. The 5250 data stream that is normally sent to the end user's display is intercepted, the display layout evaluated and translated into HTML. The browser may be:

- Windows based
- OS/2 based
- UNIX based
- Apple based

and still present the user with the AS/400 application. Thus the AS/400 system becomes platform independent.

This chapter explains some of the emulator options available if you want to access 5250 applications from the Internet. There are several products available and they solve the problems of public network communication in slightly different ways. For the purpose of this book, we have divided these products into three categories:

Category 1 Client/Server products, where both the client and the server have software modules installed.

Category 2 Web server products, where the emulator code resides on a Web server between the client and the AS/400 system.

Category 3 AS/400 based products, where the all software modules (except maybe browser plug-ins) resides on the AS/400 system.

Refer to Figure 187 for a schematic overview of the categories.

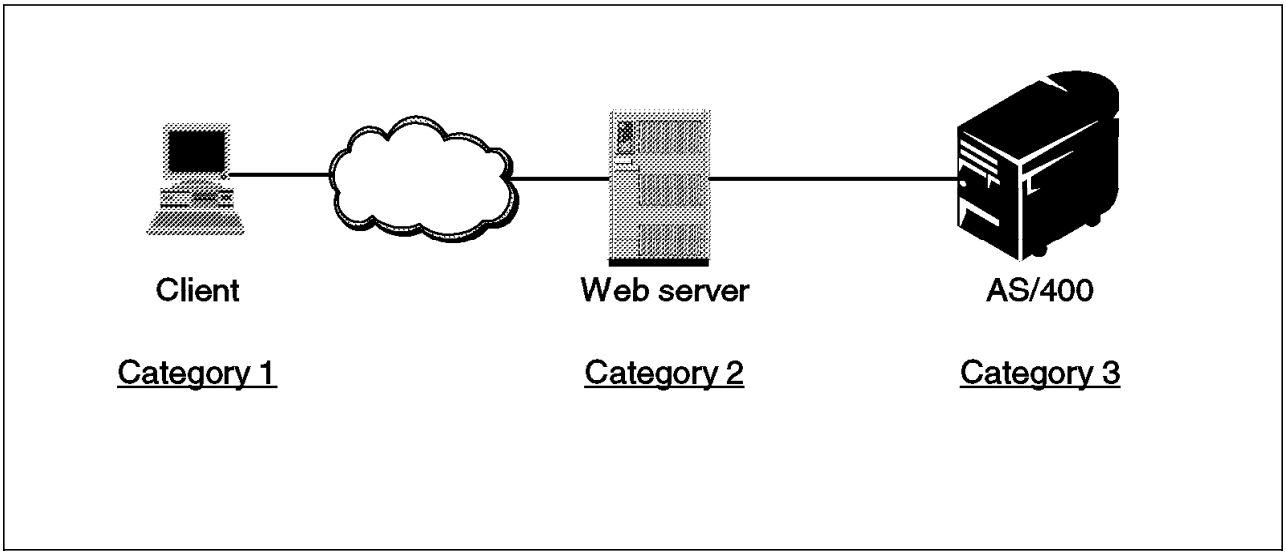


Figure 187. Emulator Products

For a more comprehensive coverage of the technicalities of emulators, see *Unleashing AS/400 Applications on the Internet*, SG24-4935.

10.1 Host On-Demand

Host On-Demand is an Internet-to-Host connectivity solution from IBM that provides host-based application discovery and access through the Internet or the corporate intranet. Web users needing access to host applications can use Host-On-Demand from inside their Java-enabled desktops or Web browsers to access central computer data. Host On-Demand uses the Java environment and TCP/IP protocols to provide the host access from within a Web browser window. By using the Java environment, Host On-Demand allows you to download the software modules to the browser, so it is in our category 2 group of products because no software other than the browser resides on the client. Host On-Demand provides these functions and benefits:

- Real 5250 ″green-screen″ emulation.
- Only requested functions are downloaded to the client.
- The keyboard and PF keys work the same as a real 5250-terminal.
- The 5250 Host On-Demand session is mouse enabled.
- Normal cut and paste functions are enabled.
- Users may remap the keyboard to their personal preferences.
- Print-screen to a locally attached printer.

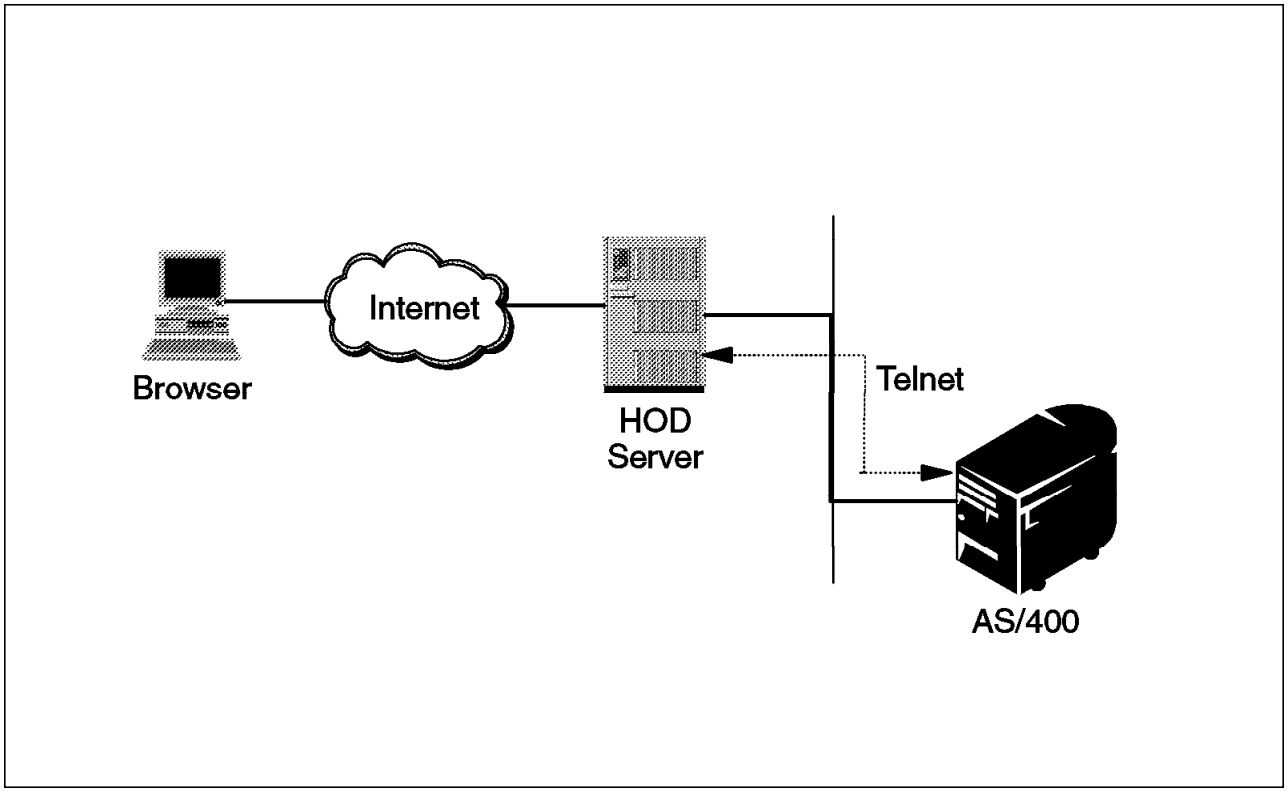


Figure 188. Host On-Demand Overview

Host On-Demand has the added benefit that SSL is supported. In the preceding environment, we can, therefore, establish a secure, encrypted, 5250 session over the Internet.

Note: The AS/400 Web server is not used in this environment.

10.1.1 Starting Host On-Demand

When the Host On-Demand (HOD) server is accessed from a Java enabled browser, the **Session selection** window is similar to Figure 189.

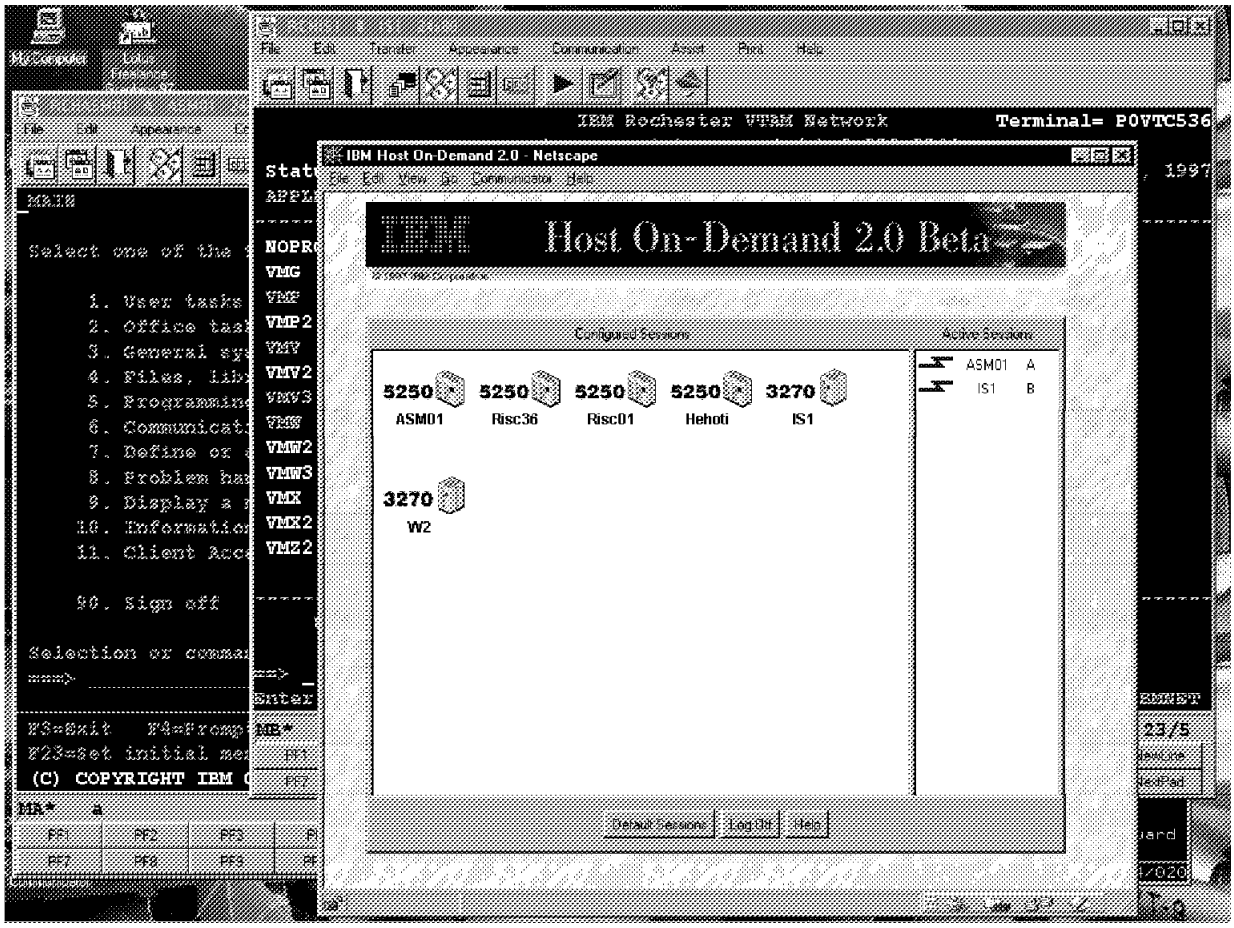


Figure 189. Host On-Demand Session Selection Window

The **Session selection** window is built up of two windows, the **Configured Sessions** window and the **Active Sessions** window. In the **Configured Sessions** window, you see all sessions that are configured for the client. The icons that represent these sessions have a **5250** or **3270** tag indicating the type of host they connect to. The **Active Sessions** window provides a list of active host sessions. This window may also be used to jump between sessions.

10.1.2 The Host On-Demand 5250 Session Window

On selecting a session from the session selection window, the Host On-Demand server presents an AS/400 Sign On display. Enter your userid and password and press Enter.

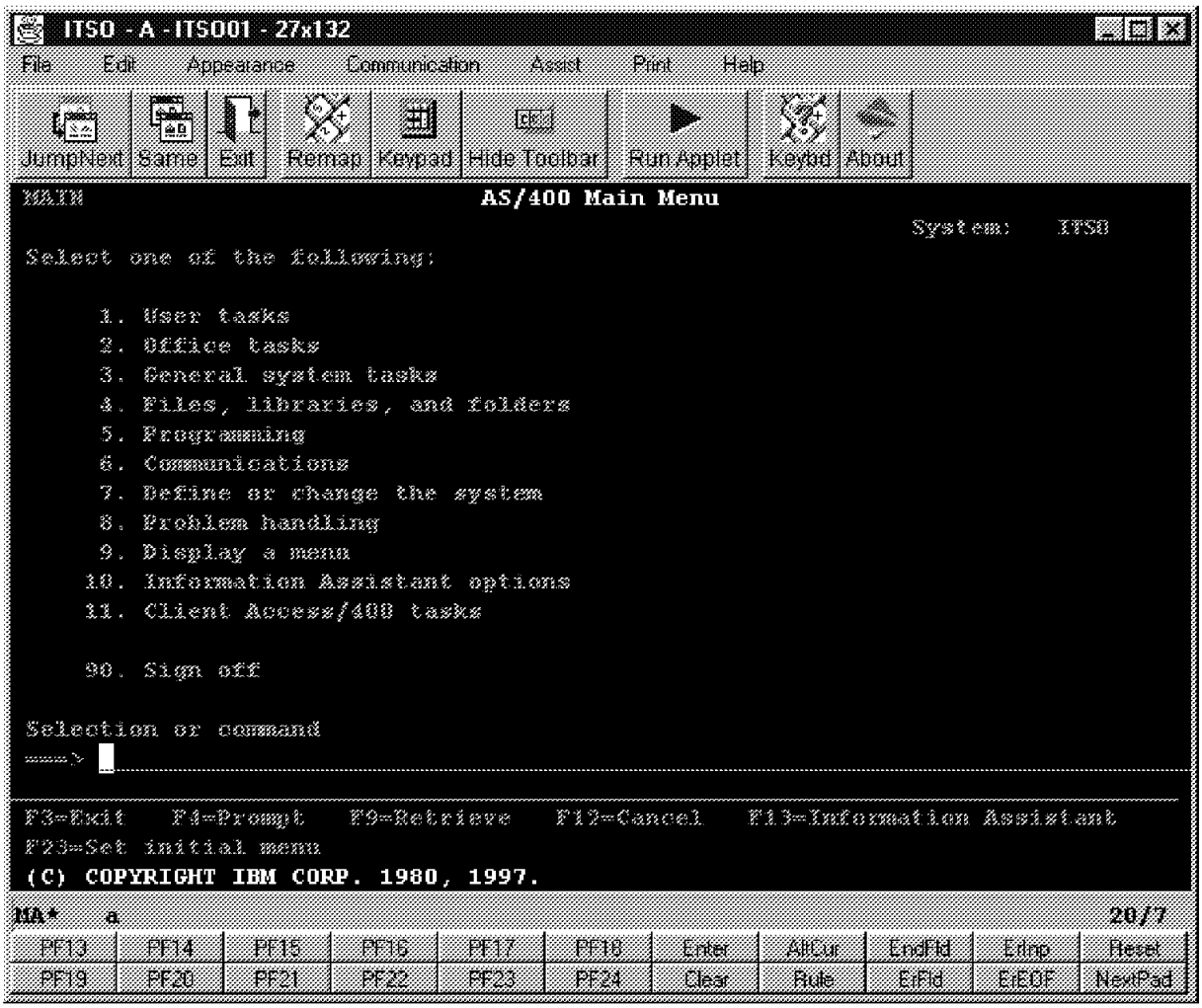


Figure 190. AS/400 Main Menu through Host On-Demand

The Host On-Demand window is divided into five logical areas:

- The **Frame** area with the AS/400 system name, the session number (letter), session name, display size, and the status bar
- The **Menu bar** area with GUI pull-down menus
- The **Tool bar** area with push buttons to productivity and configuration tools
- The **5250 screen** area from where you access the 5250 application

Note

As Host On-Demand uses the Java environment, the initial browser session remains on the Host On-Demand session selection window, and a new session is started with a **Java** icon in the upper left corner of the window.

10.2 Workstation Gateway

Workstation Gateway is an integrated part of the OS/400 TCP/IP utilities (5769-TC1) and will do on-the-fly conversion of the 5250 data stream into a HTML data stream. Workstation Gateway is a category 3 emulation product.

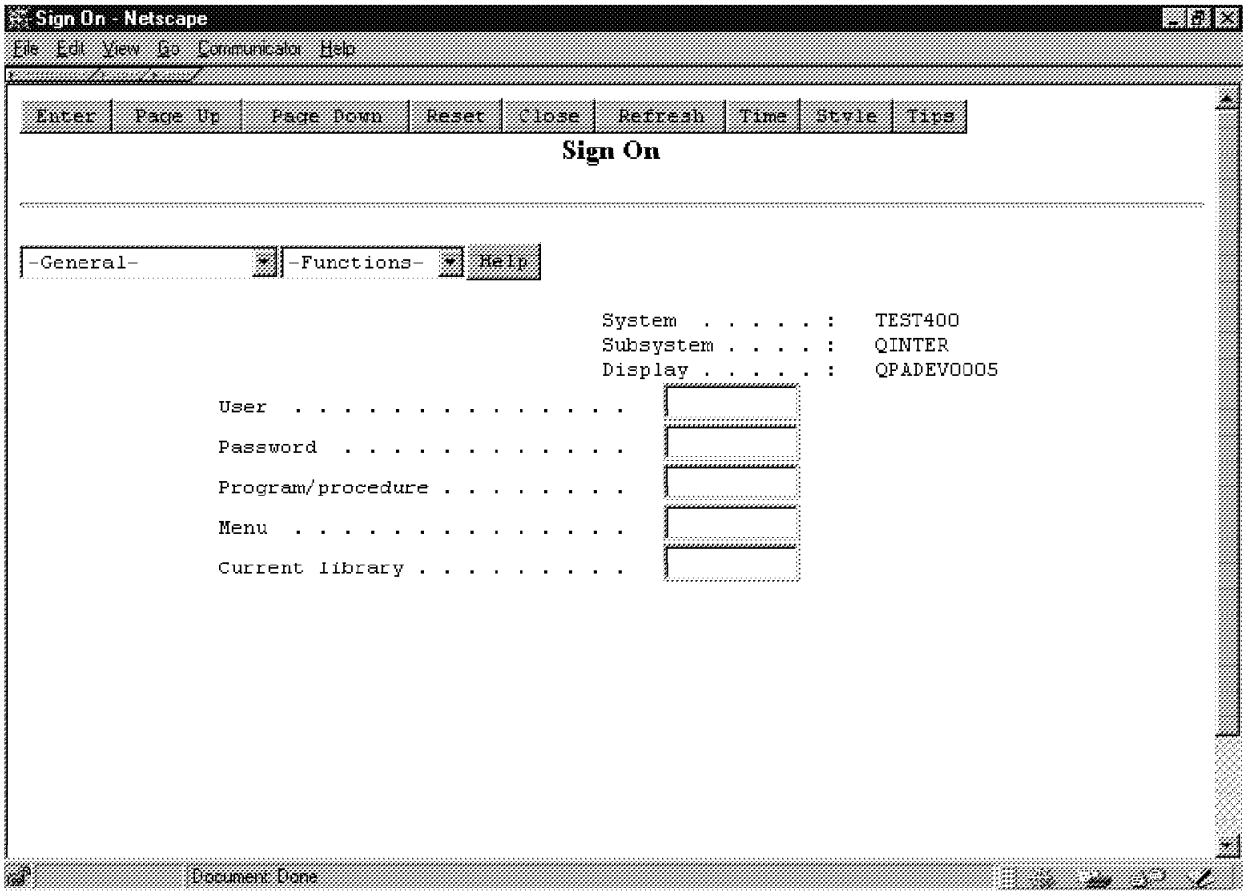


Figure 191. Workstation Gateway Sign On Display

This is the Sign On display as it looks through Workstation GateWay. The two drop-down menus provide the PF keys and special function keys required to execute 5250 applications.

10.3 Non-IBM Products

There are several ISV's supplying the market with Internet emulator products for the AS/400 system. This section *does not* intend to cover all of these products. Also remember that these products cover different needs and, hence, should be evaluated with the exact requirements in mind.

10.3.1 I/Net Webulator

Webulator is the 5250-HTML gateway from I/NET; it is used in conjunction with either Web Server/400 (provides an in-secure connection) or Commerce Server/400 (provides an SSL-encrypted session), and is a category 3 emulator product. More information about I/NET and their AS/400 Web serving products can be found at the following URL:

http://www.inetmi.com/

Webulator is a fully dynamic 5250-to-HTML gateway. It converts the AS/400 applications on the fly and provides a good graphical look and feel. A demo of the Webulator is available at:

http://www.inetmi.com/products/webulate/demo.htm

10.3.2 SEAGULL, J Walk, and GUI/400

SEAGULL has supplied the AS/400 world with GUI client tools since 1994. GUI/400 includes a 5250 emulator and, since it supports TCP/IP (TN5250), this emulator can be used over the Internet or over a corporate intranet. The new J Walk tool kit, offering both Windows and Java clients, also supports TCP/IP and Web enablement. Launched by a browser, Windows and Java J Walk clients receive a compact data stream from a central J Walk Server, resulting in efficient GUI access to AS/400 applications through the Internet. GUI/400 product is a category 1 product and J Walk is a category 2 product. Visit SEAGULL at:

http://www.seagull.nl.

10.3.3 Teubner & Associates, Inc. CORRIDOR

CORRIDOR from Teubner & Associates can be used to access any application with a 5250 interface. CORRIDOR is a category 2 product; it resides on the Web server. It translates 5250-data from the AS/400 system into HTML and HTML-data from the client into 5250 and it fully supports SSL. A good overview is available at:

http://www.teubner.com/corridor/blueprint/corrBlue.html

The following URL is an overall view of an installation with the CORRIDOR HTTP server installed.

http://www.teubner.com/corridor/blueprint/photo/corrBlueprint.html

10.3.4 Farabi Technology Corp., Hostfront

Hostfront (formerly Plexus) from Farabi Technology Corp. is a category 2 product, as it requires an NT server. It works in conjunction with Microsoft's SNA server and Web server to provide an SNA connection to the host. The product will do *some* on-the-fly conversion of the 5250 screens. SSL support can be added. The following URL takes you to Farabi Technology Corp.'s home page:

http://www.farabi.com/

10.3.5 Better On-Line Solutions, BOSaNOVA

BOSaNOVA Internet Server is Better On-Line Solution's Internet connectivity tool for the AS/400 system. It requires a Windows 95 server and the clients need special software modules, so it is a category 1/2 product. BOSaNOVA supports all APPC functions, including printing, file transfer, and EHNAPPC for client/server APIs. Better On-Line Solutions can be found at:

http://www.bosusa.com/

10.3.6 Idea, Idea Internet Host Server

Idea Internet Host Server (IHS) resides on an NT server, so it is in category 2 of emulator products. It supports SSL, does on-the-fly conversion to a graphical look and feel, and uses SNA to communicate with the host. There are also additional security features within IHS; among others a sign on mechanism that requires user authentication prior to attempting host access. Idea's home page is at:

http://www.idea.com/

10.3.7 WRQ, Reflection for TCP

WRQ develops the Reflection suite of connectivity products. The TCP Reflection suite is a category 3 product, it resides on the AS/400 system. SSL is supported and the product is Java based. Read more about WRQ and the entire Reflection product suite at:

http://www.wrq.com/

10.3.8 OpenConnect Systems, OC:Webconnect

OC://Webconnect from OpenConnect Systems is a category 2 product; it requires a Unix or NT server for the middleware. The product incorporates some excellent security features (four levels of security supported: Application, session, transport and host level). The browser must support JVM 1.02 or later. You can visit OpenConnect systems at their home page:

http://www.oc.com/

Chapter 11. National Language Support

Internet Connection Server for AS/400 and Internet Connection Secure Server for AS/400 include National Language Support (NLS). This support allows, for example, for the AS/400 system to run multiple server instances, each running a different language. In this chapter, we look at NLS on ICS for AS/400 and ICSS for AS/400, how this effects server configuration, and other matters relating to server NLS support.

National Language Support (NLS) consists of much more than translation. To really support a national language, a product must consider all aspects of the language. For example:

- The character sets that are required by the language
- The direction in which language is written
- Monetary abbreviation and symbols
- Language conversions that differ from those of U.S. English

We also have to consider that DBCS has Shift-In and Shift-Out bytes that separate the double-byte characters from single-byte characters in EBCDIC data streams.

11.1 General Considerations

The World Wide Web was originally developed based on UNIX and PC systems, which have ASCII encoding schemes. All Web related technologies were, therefore, developed with ASCII as the encoding scheme. The AS/400 system is, however, an EBCDIC based system. To serve this ASCII world as an EBCDIC Web system, the AS/400 system has to be aware of this encoding difference. In the DBCS environment, the difference is more complex.

11.1.1 Why Do We Need to Consider NLS?

If the server is only serving static HTML pages, we recommend that those pages be placed in one of the AS/400 system′s ASCII file systems. In this case, we do not have to consider NLS code conversion. But if you want to take advantage of the AS/400 system′s database or serve dynamic pages (run CGI programs), the data must be converted from EBCDIC to ASCII because database files reside in (and CGI programs run in) the QSYS.LIB file system that uses EBCDIC encoding.

In general, there are DBCS considerations on the World Wide Web, even if the HTTP server is an ASCII based system. This is because the data stream between a Web browser and a Web server is based on 7-bit ASCII, which includes single-byte characters only. This is made more complex, however, by the AS/400 system. The AS/400 system has Shift-In and Shift-Out bytes to separate double-byte characters from single-byte characters in EBCDIC data streams. PC data streams do not have these bytes; therefore, host data streams can be longer than PC data streams. We must consider these differences between client and server data streams.

Moreover, the AS/400 system has the capability to serve more than one language, potentially making code conversion even more complex. We may, for example, have Japanese CCSID files as an HTML document and serve this file based on an English CCSID before sending them to English browser. The AS/400

system has to convert the file contents (characters) from a Japanese CCSID to an English CCSID before sending them to the English browser.

11.1.2 Multiple Languages Environment

Internet Connection Server for AS/400 and Internet Connection Secure Server for AS/400 support a multiple languages environment by using multiple server instances. Each server instance can be running a different language. We can separate Web server content and applications by AS/400 National Language Version (NLV).

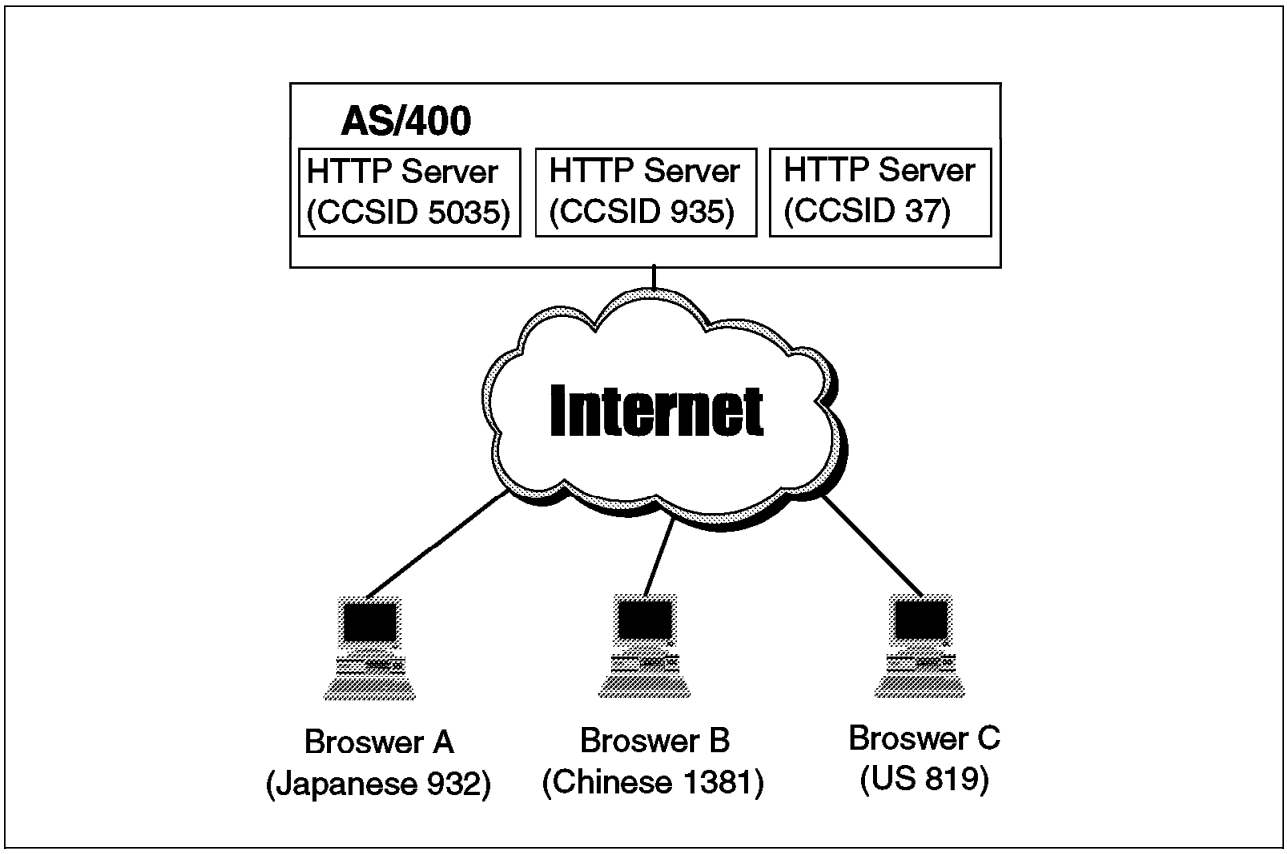


Figure 192. Multiple Server Instances for Multiple Language Environment

If serving only static ASCII HTML pages that exist in the IFS root file system, it is not necessary to consider the CCSID and code page. However, if there is a requirement to run CGI programs or for database access, we have to consider the CCSID and code page. The AS/400 QSYS.LIB file system is an EBCDIC file system. CGI programs can only run in the QSYS.LIB file system. When using the ADMIN server to create and configure server instances, we also have to consider the CCSID because the ADMIN server is an ICS for AS/400 server and it uses CGI applications.

11.2 Code Conversion Mechanisms

Let's look at the code conversion mechanism on ICS for AS/400. When serving static HTML pages from QSYS.LIB (an EBCDIC file system), the EBCDIC to ASCII code conversion performed by the server is determined by the CCSID of the file. If the HTML pages exist as ASCII files on IFS, which is the best way, no code conversion is required. When serving dynamic pages, which means running CGI programs to serve some data dynamically, the code conversion performed by the server is determined by server parameters.

11.2.1 Static Page Serving

If we want to serve only static HTML pages, the HTML documents should exist in the IFS as ASCII files. When serving ASCII HTML pages, code conversion is not necessary and this gives better performance. However, on the client side, we have to set up the browser to correctly display the page contents. For example, Netscape Navigator supports many language environments including DBCS environments. However, Netscape Navigator does not have language fonts. If you want to display DBCS characters, you must have a DBCS enabled operating system such as Windows95. Having DBCS enabled the operating system, you can then set up the browser encoding option. Figure 193 shows the Netscape Navigator encoding options.

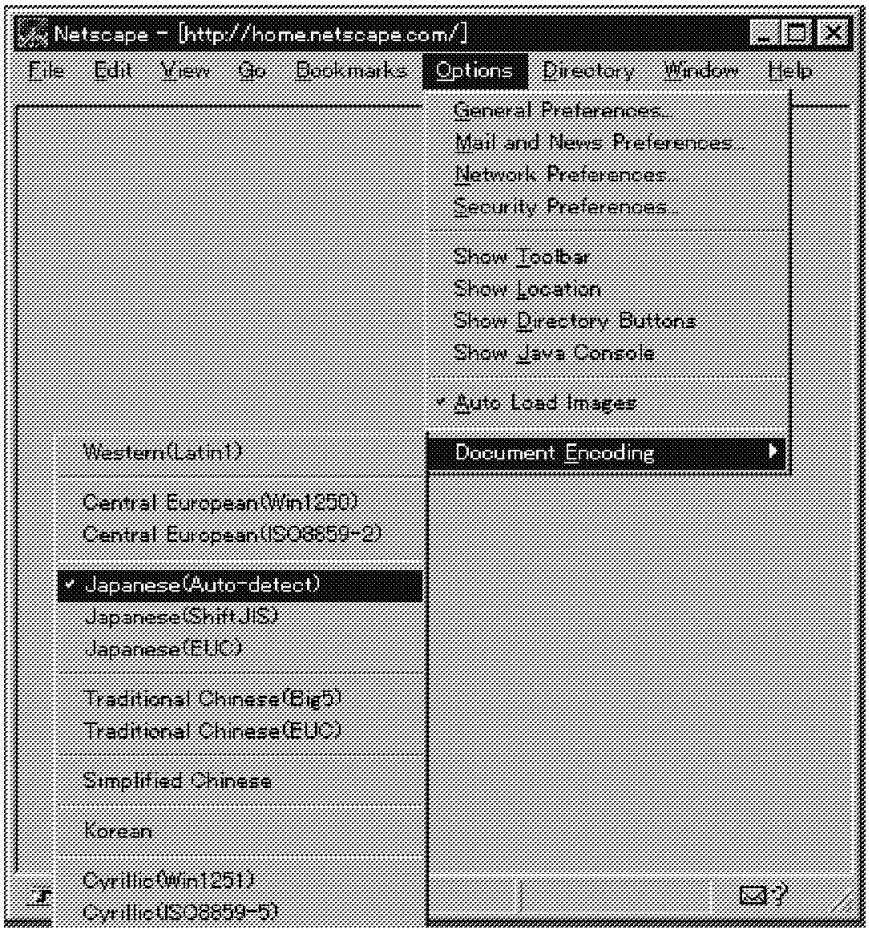


Figure 193. Netscape Document Encoding Options

If we put HTML files in QSYS.LIB, ICS for AS/400 (or ICSS for AS/400) automatically executes the EBCDIC to ASCII code conversion based on the CCSID of the file. For Net.Data, we recommend that the Net.Data macro be located in QSYS.LIB; ICS for AS/400 (or ICSS for AS/400) can then perform the EBCDIC to ASCII conversion based on the CCSID of the macro file. There are no options associated with the conversion.

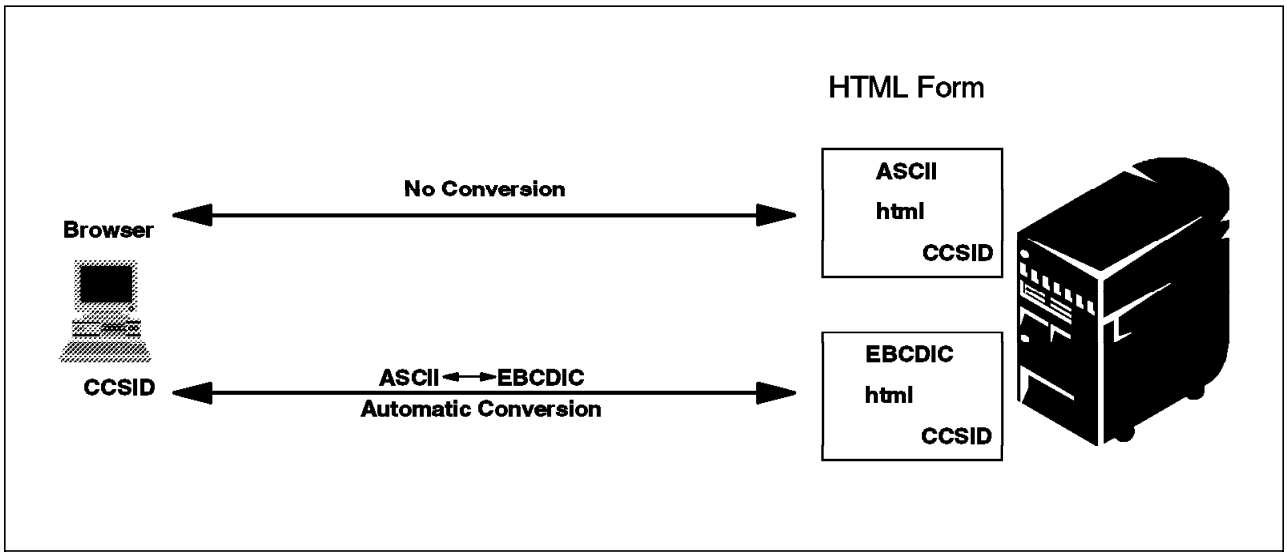


Figure 194. Code Conversion on Static Page Serving

ICS for AS/400 and ICSS for AS/400 use the following encoding schemes for the automatic code conversion.

For single byte EBCDIC CCSID files:

- Encoding scheme 4100 (single byte), ISO 8
- Encoding scheme 5100 (single byte), Japanese, Traditional Chinese, and Korean EUC single byte

For double or multi-byte EBCDIC CCSID files:

- Encoding scheme 5404 (double byte), Japanese EUC and TCP/IP; Traditional Chinese EUC on 7-bit ISO 2022; Traditional Chinese TCP/IP OS/400; ISO 2022 TCP/IP OS/400; Korean EUC on 7-bit ISO 2022
- Encoding scheme 2300 (mixed byte and double byte), IBM PC mixed byte

For example, ICS for AS/400 converts all EBCDIC files with CCSID 37 (U.S. EBCDIC) to PC code CCSID 819 (U.S. ASCII). In the case of a file with CCSID 5035 (EBCDIC Japanese-English) or 5026 (EBCDIC Japanese Katakana), all files are converted to CCSID 5052 (ISO-2020-JP Japanese InterNet Code Page extended).

In summary, when serving static HTML pages from QSYS.LIB, it is the CCSID of the HTML file that sets the code conversion. The results on the browser are dependent on this and not on the CCSID of the HTTP server job, and so on.

11.2.2 Dynamic Page Serving

Dynamic Page Serving means running CGI programs to pass information from the client to the server and back again. On the AS/400 system, CGI programs exist in QSYS.LIB which is an EBCDIC file system. So we need to convert input parameters received from Web clients to EBCDIC strings, which CGI programs can read, and convert output parameters from EBCDIC to ASCII before passing them to Web clients. There are two HTTP request form methods, GET and POST, that can be used to invoke CGI programs/scripts. The CGI program works differently depending on the method used, POST or GET, when getting the input parameters from clients. The GET method uses QUERY_STRING, which is one of the Environment Variables and the POST method uses StdIn (the standard-in file stream). This means that the code conversion mechanism used when the ASCII string is received from a Web client depends on the method used. The output from a CGI program is based on StdOut (the standard-out file stream) for both the GET and POST methods, so the code conversion mechanism from EBCDIC to ASCII is not depended on which method is used. When the GET method is being used (which uses QUERY_STRING/StdOut), we can specify code conversion parameters on output (StdOut) but not on input (QUERY_STRING). When the POST method is being used (which uses StdIn/StdOut), we can specify code conversion parameters on both input (StdIn) and output (StdOut). In the case of StdIn/StdOut, we can specify code conversion parameters in the sever configuration file or in the server instance configuration. Section 11.3.2, “Customer Server Setup” on page 218 explains how you can specify parameters for the code conversion on StdIn/StdOut. QUERY_STRING data in converted automatically with no options.

The URL-encoded data stream from a browser is based on 7-bit ASCII. Some special characters, Katakana, and double-byte characters are encoded into an escape sequence in hexadecimal format (%2c as an example). The AS/400 CGI interface performs a CCSID819 to CCSID37 conversion on this URL-encoded data and stores the result in a QUERY_STRING with a CCSID designation of 37. We do not, therefore, need to code conversion parameters but we do need to take this into consideration on CGI programming when parsing and decoding.

Note: If you need more information about CGI and Net.Data, refer to *Unleashing AS/400 Applications on the Internet*, SG24-4935.

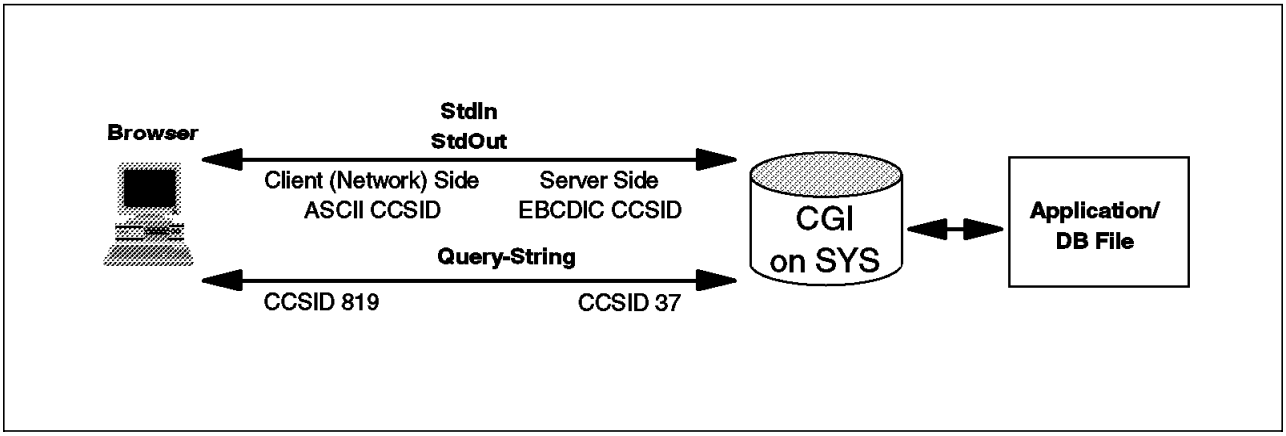


Figure 195. Code Conversion on Dynamic Page Serving

11.3 Server Configuration for NLS

ICS for AS/400 and ICSS for AS/400 have five server directives for NLS. You can specify CCSID values by using these directives. The five server directives for NLS are:

- **AddLanguage:** Specify the language of files depending on the file extension.
- **DefaultFsCCSID:** Specify server character set environment.
- **DefaultNetCCSID:** Specify client character set environment.
- **TblHttpIn:** Specify the incoming ASCII to EBCDIC conversion table.
- **TblHttpOut:** Specify the outgoing EBCDIC to ASCII conversion table.

Refer to the *Webmaster's Guide*, GC41-5434, for more information on these directives.

11.3.1 ADMIN Server Setup

The ADMIN server can be configured such that it uses an installed NLV. Once the ADMIN server is set up, the customer's ICS for AS/400 or ICSS for AS/400 servers can be configured using that language. If you want to to run the ADMIN server in a secondary language, that language must be installed for the TC1 product. The AS/400 user profile used to access the ADMIN server must have the LANGID value set to the secondary language. The ADMIN server windows then use the secondary language. If the user profile language is not installed for TC1, the server uses English.

The ADMIN server interface accesses CGI programs. We, therefore, need to set the server and client CCSID values. If we do not do this, the server uses the QCCSID system value for the server CCSID and the default ASCII CCSID of 819. The server CCSID can be set using the **-fsccsid** parameter of the STRTCPSVR command:

```
STRTCPSVR SERVER(*HTTP) HTTPSVR(*ADMIN '-FSCCSID xxx')
```

Where *xxx* is the required EBCDIC CCSID. The client side CCSID can be set using the CHGHTTPA command:

```
CHGHTTPA CCSID(yyy)
```

Where *yyy* is the required ASCII CCSID.

11.3.2 Customer Server Setup

We can use separate server instances to handle each NLV and configure each instance for the specific NLV. When configuring a server instance for NLV, we specify the server side EBCDIC CCSID and the client side ASCII CCSID. These CCSIDs are then used by the StdIn/StdOut of CGI programs for code conversion.

11.3.2.1 Server Side EBCDIC CCSID Setup

The server side EBCDIC CCSID is used as the CCSID of the server job and for code conversion on StdIn/Out. The server side EBCDIC CCSID used is determined from the following parameters in this order of priority:

1. **-fsccsid** server startup value
2. **DefaultFsCCSID** server directive
3. **QCCSID** system value

If the *-fsccsid* server startup value is used when starting a server instance, the server job works under this CCSID, which is the EBCDIC CCSID used for code conversion. If this parameter is not specified, the CCSID specified in the server configuration, *DefaultFsCCSID* server directive, is used. If the server configuration does not have a *DefaultFsCCSID* server directive, the server uses the CCSID specified as the *QCCSID* system value.

-fsccsid server startup value

The *-fsccsid* server startup value is set using the STRTCPSVR command:

```
STRTCPSVR SERVER(*HTTP) HTTPSVR(instname '-FSCCSID xxxx')
```

Where *instname* is your server instance name and *xxxx* is the EBCDIC CCSID you want.

DefaultFsCCSID server directive

The *DefaultFsCCSID* server directive is set using the WRKHTTPCFG command:

```
                         Work with HTTP Configuration
                                                              System:   HOST01
   Configuration name . . . . . . . :   SAORI01

 Type options, press Enter.
   1=Add   2=Change   3=Copy   4=Remove   5=Display   13=Insert

       Sequence
Opt   Number    Entry

1_  00100     DefaultFsCCSID 5035
__  00010     ##############################################
__  00020     #            HTTP SERVER CONFIGURATION
__  00030     ##############################################
__  00040     HostName host01.domain01.ibm.com
__  00050     Port    6101
__  00060       Enable               GET
__  00070       Enable               POST
__  00080       Enable               HEAD
                                                                        More...
 F3=Exit   F5=Refresh   F6=Print List   F12=Cancel   F17=Top   F18=Bottom
 F19=Edit Sequence
```

Figure 196. Using the WRKHTTPCFG Command to Add the DefaultFsCCSID Directive

In the preceding example, we have set the DefaultFsCCSID value to 5035.

QCCSID system value

The *QCCSID* system value is set using the CHGSYSVAL command:

```
CHGSYSVAL SYSVAL(QCCSID) VALUE(xxxx)
```

Where *xxxx* is the EBCDIC CCSID you want.

Note: Please be aware that changing this value will effect other jobs.

11.3.2.2 Client Side ASCII CCSID Setup

The Client ASCII CCSID is used on the network side for code conversion on StdIn/Out. The client side ASCII CCSID used is determined from the following parameters in this order of priority:

1. **-netccsid** server startup value
2. **DefaultNetCCSID** server directive
3. **CCSID** value on HTTP server attribute

If the *-netccsid* server startup value is used when starting a server instance, the ASCII CCSID used for code conversion is this value. If this parameter is not specified, the CCSID specified in the server configuration, *DefaultNetCCSID* server directive, is used. If the server configuration does not have a *DefaultNetCCSID* server directive, the server uses the CCSID specified as the *CCSID* HTTP server attribute.

-netccsid server startup value

The *-netccsid* server startup value is set using the STRTCPSVR command:

```
STRTCPSVR SERVER(*HTTP) HTTPSVR(instname '-NETCCSID xxxx')
```

Where *instname* is your server instance name and *xxxx* is the ASCII CCSID you want.

DefaultNetCCSID server directive

The *DefaultNetCCSID* server directive is set using the WRKHTTPCFG command:

```
                         Work with HTTP Configuration
                                                              System:   HOST01
   Configuration name . . . . . . . :   SAORI01

 Type options, press Enter.
   1=Add   2=Change   3=Copy   4=Remove   5=Display   13=Insert

       Sequence
Opt   Number    Entry

1_  00100     DefaultNetCCSID 932
__  00010     ################################################
__  00020     #            HTTP SERVER CONFIGURATION
__  00030     ################################################
__  00040     HostName host01.domain01.ibm.com
__  00050     Port    6101
__  00060       Enable               GET
__  00070       Enable               POST
__  00080       Enable               HEAD
                                                                        More...
 F3=Exit   F5=Refresh   F6=Print List   F12=Cancel   F17=Top   F18=Bottom
 F19=Edit Sequence
```

Figure 197. Using the WRKHTTPCFG Command to Add the DefaultNetCCSID Directive

In the preceding example, we have set the DefaultNetCCSID value to 932.

CCSID server attribute

The *CCSID* server attribute is set using the Global Attribute Values form from the ADMIN server General Configuration and

Administration form (see Figure 33 on page 52) or using the CHGHTTPA command:

```
                      Change HTTP Attributes (CHGHTTPA)

Type choices, press Enter.

Autostart  . . . . . . . . . . . AUTOSTART      *NO
Number of server jobs:          NBRSVR
  Minimum  . . . . . . . . . .                  3
  Maximum  . . . . . . . . . .                  15
Coded character set identifier  CCSID          00819
Server mapping tables:          TBLHTTPOUT
  Outgoing EBCDIC/ASCII table  .                *CCSID
    Library  . . . . . . . . .                    _______
                                TBLHTTPIN
  Incoming ASCII/EBCDIC table  .                *CCSID
    Library  . . . . . . . . .                    _______




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

Figure 198. Change HTTP Attributes (CHGHTTPA)

In the preceding example, we have set the CCSID server attribute to 00819.

Note: The HTTP server attribute CCSID value sets the CCSID for *all* HTTP servers. For an individual server, use the DefaultNetCCSID server configuration directive or the -NetCCSID server startup parameter.

Chapter 12. Building an Internet Server Site

There are numerous advantages to be gained from connecting an AS/400 system to the Internet as a Web server. This may, for example, be to display a product catalog to the World using static Web pages on a stand-alone Web server or; it may be a fully-interactive Web site where Web clients can purchase goods and services, and the information on the server is dynamically updated from a production AS/400 system.

Whatever the intentions for the Internet Connection Server for AS/400, there are security considerations that should be addressed before connecting the server and the company network to the Internet.

Although it is outside the scope of this redbook to provide a complete guide to connecting a Web server to the Internet such that the server is protected from harm, we briefly discuss some of the following areas so that you have a basic understanding of the issues:

- Connecting to an intranet or the Internet
- Internet Service Providers (ISP)
- Security and the Internet
- Firewalls

12.1 Connecting to an Intranet or the Internet

You can connect a Web server to a corporate intranet, to the Internet, or both. If a Web server is only connected to an intranet with no external Internet connections, this procedure can be carried out by the company network administrator. The security exposures should be less, although still present. Functions such as access control and OS/400′s excellent native security features provide a high degree of security against attacks by malicious insiders such as a disgruntled employee. If, however, a Web server and a corporate intranet is connected to the Internet, this raises some important questions such as:

- Should internal users be able to access Internet resources?
- Should external Internet users have access to the company Web server?
- Which applications should be accessible by Internet users?
- Does data on the server need to be constantly updated?
- How should the internal network and server be protected?

First of all though, there should be a clear idea of what the company is trying to achieve by connecting to the Internet. Planning at this stage is vitally important. By establishing a Web presence you are putting the corporate image on view to millions of people 24 hours a day. If the Web site is badly planned, hard to use, or just looks bad, this reflects on the company. Remember though, with some planning, it is possible for even a tiny business to implement a Web site every bit as professional looking as a multi-national corporation. If the Web site looks similar to a multi-national corporation, that is how the company is perceived.

Assuming that you have a clear plan, maybe arrived at with the help of expert consultants, you now need to physically attach the Web server and internal network to the Internet.

To connect the server and internal network to the Internet, you need the services of an Internet Service Provider (ISP). There are many reputable ISPs such as IBM Global Networks (IGN) who can perform this service for you.

12.2 Internet Service Providers

An Internet Service Provider (ISP) provides a connection to the Internet for you. Typically this connection includes a leased line to the premises and a router to connect the internal Local Area Network (LAN) to the leased line. An ISP usually has high-speed links into the Internet to which you can gain access to using the leased line. It is also possible to establish a dial-up link to an ISP but, taking into account that it is desirable for a Web site to be available 24 hours a day, a permanent leased line connection is more usual.

Figure 199 shows how a company, called Mycompany, connected their intranet to the Internet. The ISP, which has several links into the Internet, provided a router at the customer premises that connects with the router at the ISP premises through a leased line and then to the Internet. This is a high-level diagram. It does not represent a desirable solution as no security measures have been implemented.

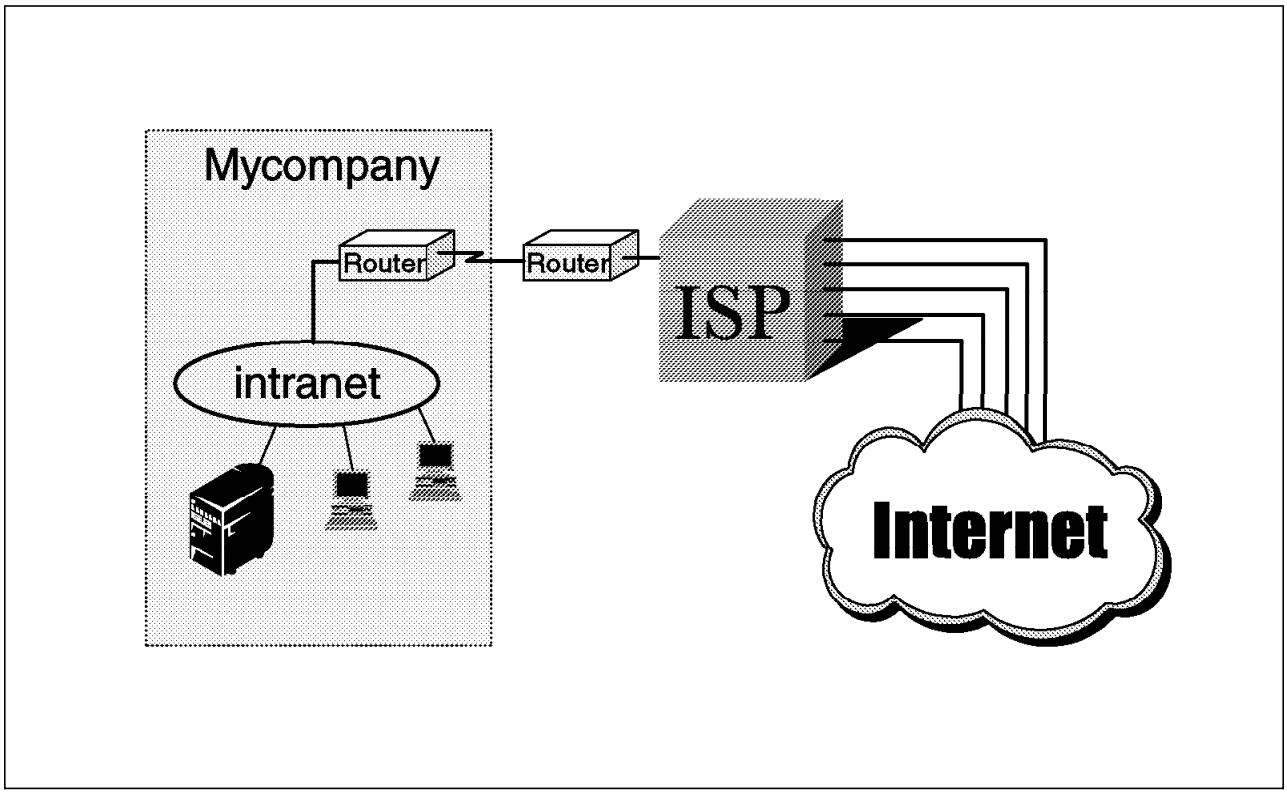


Figure 199. Connecting to the Internet through an ISP

Most reputable ISPs also handle other functions involved when connecting to the Internet; these may include:

- Provision of unique IP addresses for the company
- Provision of a public domain name for the company
- Provision of Domain Name Services
- Maintenance of the router at the premises
- Consultancy in planning, design, and setup of an Internet connection

For a full list of functions and services provided by an ISP, contact them directly.

Once an ISP has connected the LAN to the Internet, it is possible, in theory, for any user on the intranet using TCP/IP protocols and applications to communicate with any similar user anywhere on the Internet. It is also possible for any user on the Internet to communicate with users and systems, including Web servers, on the LAN. Here lies the main concern, security. The goal should be to implement those aspects of the Internet that suit the business needs without the risks associated when connecting the network to the Internet.

12.3 Security and the Internet

Protecting the internal systems against attack from the Internet should be the prime concern. **Always** consider the implications of any action you undertake on the Internet such as allowing access to the Web server. Internet security is about the implementation of security policies.

A security policy can be as simple as saying the Web server will not be connected to the intranet and will only serve static Web pages to Internet clients. This bypasses a lot of security issues and is easy to implement. Another, more complicated but realistic security policy can take into account:

System security
: The AS/400 system offers a strong set of security tools that should be considered such as user profile and password management and system-wide security values.

Physical security
: It is no use taking trouble to secure the AS/400 system if any employee can walk up to it and power it off or unplug communication cables. Badge-controlled doors may be part of a security policy.

Application security
: Every TCP/IP application has a different inherent security risk. TELNET has a high inherent security risk and so the policy may state that the TELNET server is not to be active on the server.

Transaction security
: Commercial transactions through the Internet should be protected. You may decide to enforce encryption of all data and authentication of clients.

Network security
: Access to the corporation's network should be limited. To what extent it is limited should be laid down in a security policy.

Without documented security policies, do not proceed to connect the Web server and internal network to the Internet. The threat from the Internet is serious and a real one; no company should connect to the Internet without understanding the

risks involved. We strongly recommend that anyone wanting to connect to the Internet read the redbook *AS/400 Internet Security: Protecting your AS/400 from HARM on the Internet*, SG24-4929, and consult with experts in the area such as the ISP or a specialist Internet security company. There are an estimated 30 million people with Internet access; ask yourself whether you want any or all of them to have access to the corporate network and business systems.

There is a fundamental conflict between security and access, each requiring a trade-off with the other, more security means less access and vice-versa. With each new connection or new TCP/IP application you provide for the users and customers, you have given a potential intruder one more means to enter the system. It should be clear that there is always a risk with being attached to the Internet. However, the benefits for the company being present in the Internet are many. But, it is a high-level management decision whether and how to deal with the Internet and to consider the risks. These policies should be part of the overall I/T and networking policies and strategies. So, before connecting a system to the Internet, you need to make explicit decisions on what type of access and service is offered.

One point we should make concerning security is that this is not just an AS/400 issue, this is a problem for any server or system on the Internet. The AS/400 system has extremely strong security features; however, as with all security features, if you do not use them, they are of no value. You must adopt a mind-set that is basically untrusting. For example, assume you have malicious employees, do not trust the IP address of Web clients and do not assume that the security policies are effective forever. Some employees may decide to corrupt data on the server, a faked IP address may fool access control, and hacker′s techniques evolve and change with time.

The Internet security policy should be in the context of the larger I/T security policy. It needs to be determined before you look at individual solutions and it should be determined by higher-level management.

One of the more common requirements of an Internet security policy is the inclusion of a firewall. A firewall is not a total security solution; it implements network security. One aspect of an Internet security policy is that it helps restrict access to the corporate network. It may help to understand the basic principles of a firewall.

12.4 Firewalls

A firewall acts as a **chokepoint** through which all traffic to and from the Internet flows. It prevents unwanted Internet traffic from entering the secure network, while selectively allowing internal users access to the Internet. Security weaknesses on client and server systems in the internal network cannot be exploited from the Internet when a firewall is in place. Internal users can access the servers on the Internet and exchange mail with other Internet users through the firewall, while other types of TCP/IP access from the Internet can be selectively or entirely blocked.

A firewall concentrates security administration, enforcing security policy and minimizing the opportunity for security configuration errors. It provides privacy by preventing internal network information from being accessed through the Internet. An important feature of a firewall is logging, which enables network use and misuse to be monitored. A firewall provides flexible configuration,

enabling support for various security policies. The administrator decides which services should be permitted and which should be blocked.

Firewall for AS/400 (5769-FW1) is one such product based on an Integrated PC Server that provides processor separation without requiring the introduction of a new system into the network. See *AS/400 Internet Security: IBM Firewall for AS/400*, SG24-2162, for more information on this product and on firewall implementation in general.

Remember, a firewall is not simply a box that can be switched on to provide complete protection. It is a combination of configurable technologies, software and hardware that can provide as much or as little protection as the security policy demands.

A firewall and a Web server do not necessarily go hand in hand though. Depending on how you connect the Web server and internal network to the Internet, there may or may not be a requirement for a firewall.

There are several different ways of implementing the AS/400 system as a Web server:

- Isolated Internet server
- Integrated Internet server
- Intranet server

There may be various flavors of these basic types but these three allow us to look at some of the issues, advantages, and disadvantages specific to each environment.

The Isolated Internet Server: This does not pose any threat to the company's internal network. The Web server is directly connected to the ISP using a router and an isolated LAN. If the AS/400 Web server is compromised, it does not have any adverse effect upon the company's internal network as there is no connection between the two. It is essential that no link is established between the server and any system on the intranet while the link to the Internet is operational. This can be inconvenient in terms of updating information on the Web server. Also, internal users cannot access the server. This can, however, be an ideal configuration to use as a starting point due to the low security risk. Figure 200 shows an example of an isolated, or stand-alone, Web server.

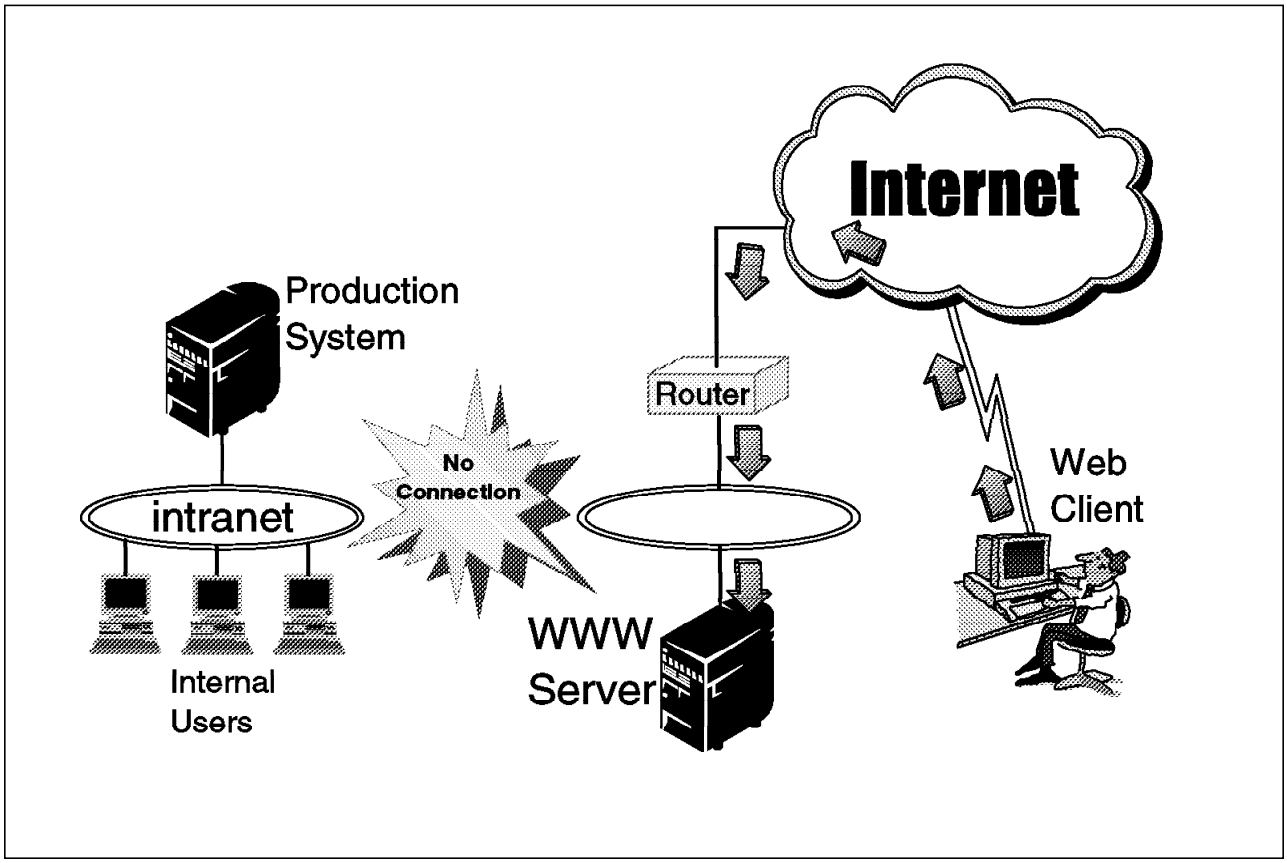


Figure 200. Isolated Internet Web Server

The Integrated Internet Server: You may want a better way of maintaining the data on the Web server or, you may want to provide the internal users with Internet access. In this case, you may want to consider connecting the internal network to the Web server and the Internet. Figure 201 and Figure 202 on page 230 show two examples of how this can be achieved securely with the use of a firewall. In the first example shown in Figure 201, the Web server is in front of the firewall. In this configuration, the Web server is attached to an unsecure LAN. This LAN is sometimes called a perimeter network or DMZ (Demilitarized Zone). The Web server may have some protection from a router that performs packet filtering but it is not protected by the firewall. The Web server is intended to be accessed by users on the Internet so full firewall protection is not possible or necessary. The advantage of this configuration is that traffic from the Internet does not need to go through the firewall, thus reducing response time and the load on the firewall. Also, more importantly, if the Web server is compromised, the internal network is still protected by the firewall. It is possible to transfer data between server and production systems. This is useful for backup and recovery as well as for use in a live Web application, where information must be up-to-date such as pricing information. It can be seen that internal users can access the Internet and the server through the firewall.

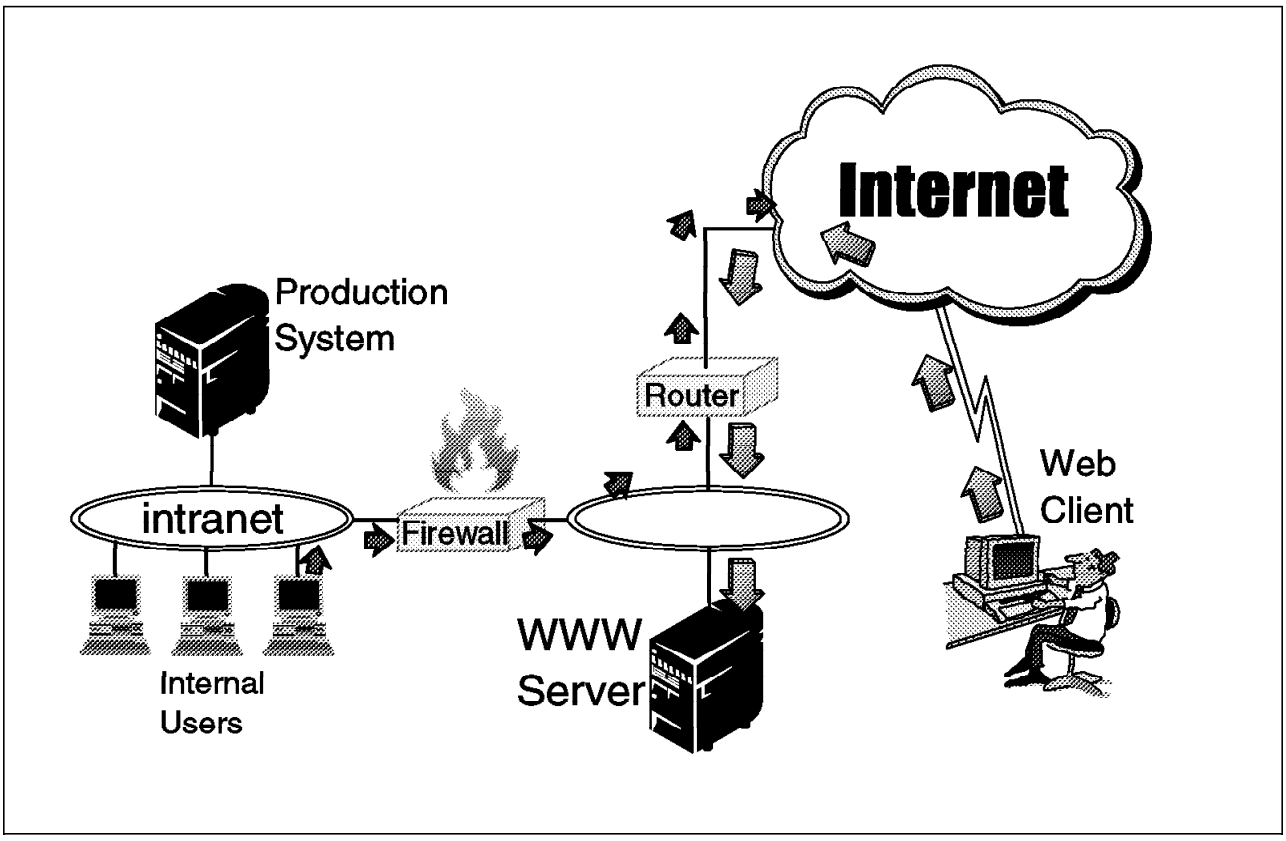


Figure 201. Server In Front of Firewall

In Figure 202, the Web server is behind the firewall, connected to the corporate intranet. In this configuration, requests from the Internet destined for the Web server must actually flow on the corporate LAN. This solution has the advantage of making updates between Web server and the production system much easier and allows fast response to internal HTTP requests but has the drawback that the internal LAN is visible to Internet users. However, the router and firewall can be used to ensure Internet traffic is only routed to the Web server.

This solution is widely used; however, if the Web server is compromised, the corporate network is at risk and, therefore, requires careful planning and setup.

It can be seen that Internet traffic must flow on the intranet to reach the server. Internal users can still access the Internet through the firewall.

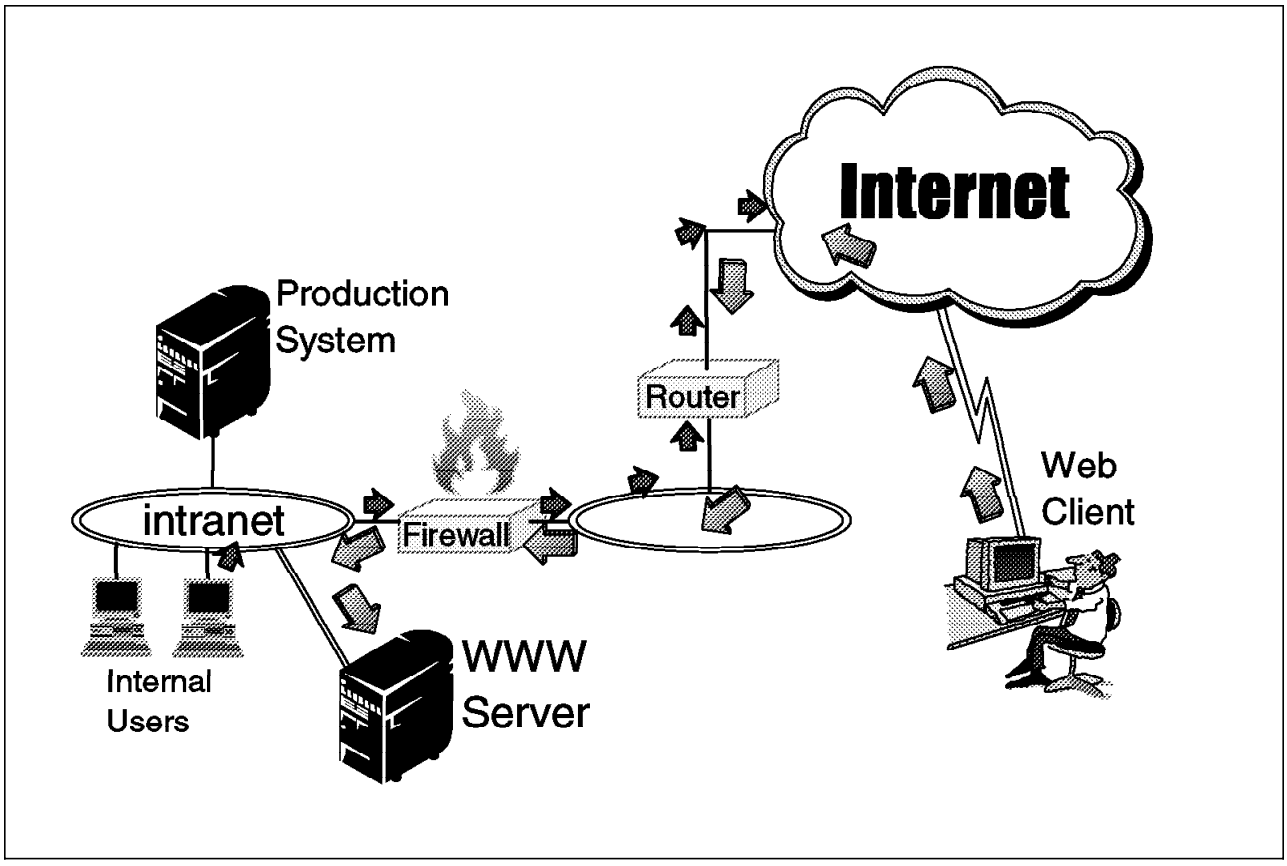


Figure 202. Server Behind Firewall

The Intranet Server: You can use the functions provided by Internet Connection Server for AS/400 to service the internal users. In this case, you are using the same applications but you are not connecting to the Internet and, therefore, the security considerations are the same that apply to any privately owned network.

Figure 203 shows an example of an intranet server. Internal users can access all the functions of Internet Connection Server for AS/400 but there is no connection to the Internet.

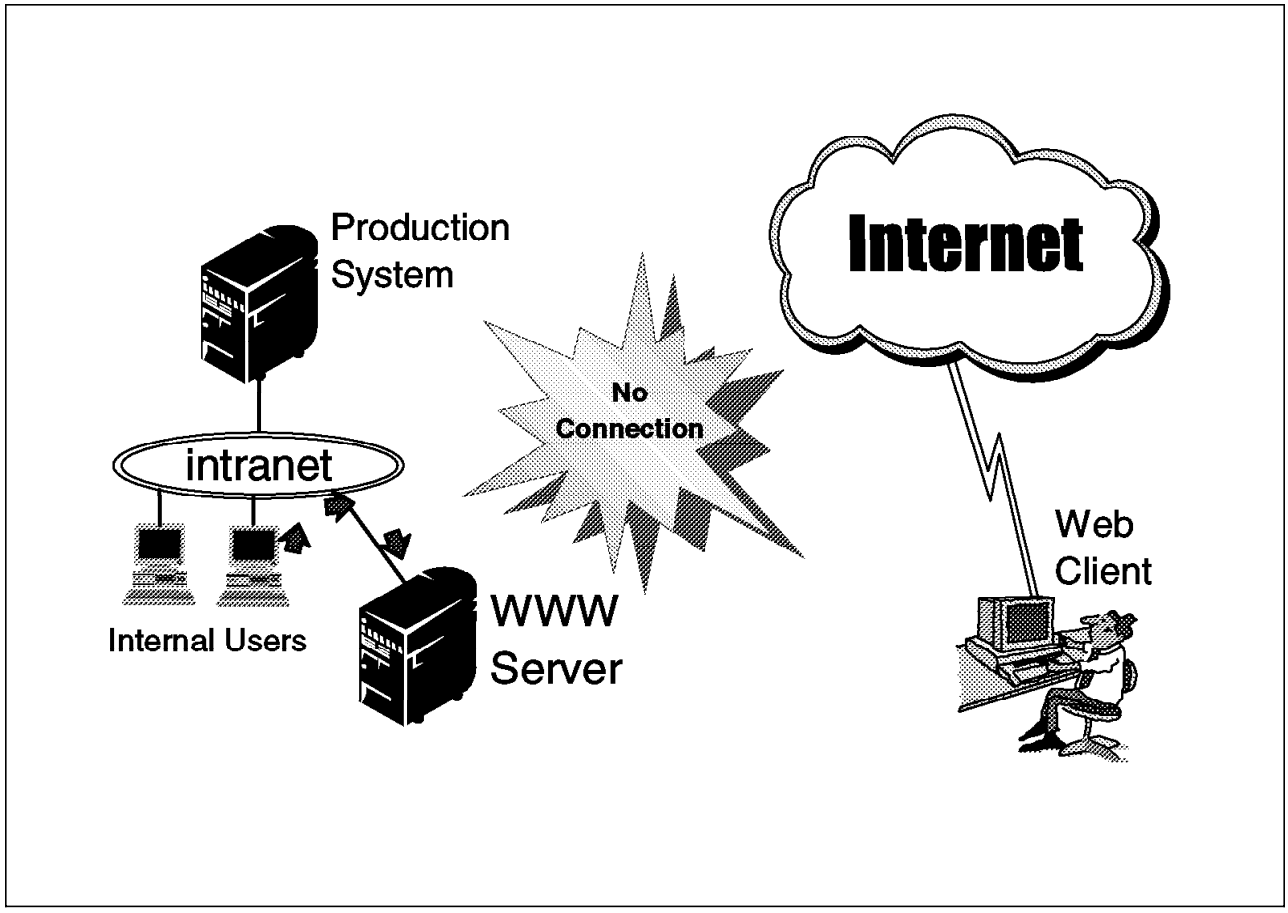


Figure 203. Intranet Server

This is by no means the definitive list. There are so many different scenarios that it is impossible to document them all. The solution that is best for the company depends on factors such as:

- The layout of the existing network
- Requirements of internal users
- Desired sophistication of the Web site and Web applications
- Budget constraints
- Technical expertise

These are the reasons why initial planning is so crucial, it is more preferable to implement a *perfect* solution from scratch, as recommended by an Internet security consultant, than to try and fix a poor solution that evolved without planning.

Appendix A. Special Notices

This publication is intended to help AS/400 technical specialists to understand and configure Internet Connection Server for AS/400 and Internet Connection Secure Server for AS/400. The information in this publication is not intended as the specification of any programming interfaces that are provided by Operating System/400 (OS/400), TCP/IP Connectivity Utilities for AS/400, or Internet Connection Secure Server for AS/400 product names 5769-SS1, 5769-TC1, 5769-NC1, and 5769-NCE. See the PUBLICATIONS section of the IBM Programming Announcement for Operating System/400 (OS/400), TCP/IP Connectivity Utilities for AS/400, or Internet Connection Secure Server for AS/400 product names 5769-SS1, 5769-TC1, 5769-NC1, and 5769-NCE for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM′s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM (″vendor″) products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

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

AIX®
Client Access
IBM®
Net.Data
OS/2®
OS/400®
400®
AS/400®
DB2®
Language Environment®
Operating System/400®
OS/390
S/390®

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

Java and HotJava are trademarks of Sun Microsystems, Incorporated.

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

Pentium, MMX, ProShare, LANDesk, and ActionMedia are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

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

Appendix B. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

B.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see “How to Get ITSO Redbooks” on page 237.

- *Unleashing AS/400 Applications on the Internet*, SG24-4935
- *AS/400 Internet Security : Protecting your AS/400 from HARM on the Internet*, SG24-4929
- *AS/400 Internet Security: IBM Firewall for AS/400*, SG24-2162 (available soon)
- *AS/400 e-commerce: Net.Commerce*, SG24-2129 (available soon)

B.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	Subscription Number	Collection Kit Number
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
Lotus Redbooks Collection	SBOF-6899	SK2T-8039
Tivoli Redbooks Collection	SBOF-6898	SK2T-8044
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RS/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RS/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
RS/6000 Redbooks Collection (PDF Format)	SBOF-8700	SK2T-8043
Application Development Redbooks Collection	SBOF-7290	SK2T-8037

B.3 Other Publications

These publications are also relevant as further information sources:

- *TCP/IP Configuration and Reference*, SC41-5420
- *Integrated File System Introduction*, SC41-5711
- *ICS, ICSS Webmaster's Guide*, GC41-5434-01
- *AS/400 and the Internet*, G325-6321

How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at http://www.redbooks.ibm.com/.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **PUBORDER** — to order hardcopies in United States
- **GOPHER link to the Internet** - type GOPHER.WTSCPOK.ITSO.IBM.COM
- **Tools disks**

  To get LIST3820s of redbooks, type one of the following commands:

  ```
  TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
  TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
  ```

  To get BookManager BOOKs of redbooks, type the following command:

  ```
  TOOLCAT REDBOOKS
  ```

  To get lists of redbooks, type the following command:

  ```
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
  ```

  To register for information on workshops, residencies, and redbooks, type the following command:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1998
  ```

  For a list of product area specialists in the ITSO: type the following command:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
  ```

- **Redbooks Web Site on the World Wide Web**

  http://w3.itso.ibm.com/redbooks/

- **IBM Direct Publications Catalog on the World Wide Web**

  http://www.elink.ibmlink.ibm.com/pbl/pbl

  IBM employees may obtain LIST3820s of redbooks from this page.

- **REDBOOKS category on INEWS**
- **Online** — send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL
- **Internet Listserver**

  With an Internet e-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an e-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

Redpieces

For information so current it is still in the process of being written, look at "Redpieces" on the Redbooks Web Site (http://www.redbooks.ibm.com/redpieces.html). Redpieces are redbooks in progress; not all redbooks become redpieces, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** — send orders to:

	IBMMAIL	**Internet**
In United States:	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada:	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America:	dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** — send orders to:

IBM Publications	IBM Publications	IBM Direct Services
Publications Customer Support	144-4th Avenue, S.W.	Sortemosevej 21
P.O. Box 29570	Calgary, Alberta T2P 3N5	DK-3450 Allerød
Raleigh, NC 27626-0570	Canada	Denmark
USA		

- **Fax** — send orders to:

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

- **1-800-IBM-4FAX (United States)** or **(+1)001-408-256-5422 (Outside USA)** — ask for:

  Index # 4421 Abstracts of new redbooks
  Index # 4422 IBM redbooks
  Index # 4420 Redbooks for last six months

- **Direct Services** - send note to softwareshop@vnet.ibm.com
- **On the World Wide Web**

Redbooks Web Site	http://www.redbooks.ibm.com/
IBM Direct Publications Catalog	http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Internet Listserver**

  With an Internet e-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an e-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank).

Redpieces

For information so current it is still in the process of being written, look at "Redpieces" on the Redbooks Web Site (http://www.redbooks.ibm.com/redpieces.html). Redpieces are redbooks in progress; not all redbooks become redpieces, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

IBM Redbook Order Form

Please send me the following:

Title	Order Number	Quantity

First name Last name

Company

Address

City Postal code Country

Telephone number Telefax number VAT number

- Invoice to customer number
- Credit card number

Credit card expiration date Card issued to Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

Index

Special Characters

Numerics

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

ITSO Redbook Evaluation

AS/400 e-commerce: Internet Connection Servers
SG24-2150-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at http://www.redbooks.ibm.com
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction _____

Please answer the following questions:

Was this redbook published in time for your needs? Yes____ No____

If no, please explain:

_____

_____

_____

_____

What other redbooks would you like to see published?

_____

_____

_____

Comments/Suggestions: **(THANK YOU FOR YOUR FEEDBACK!)**

_____

_____

_____

_____

_____

SG24-2150-00
Printed in the U.S.A.