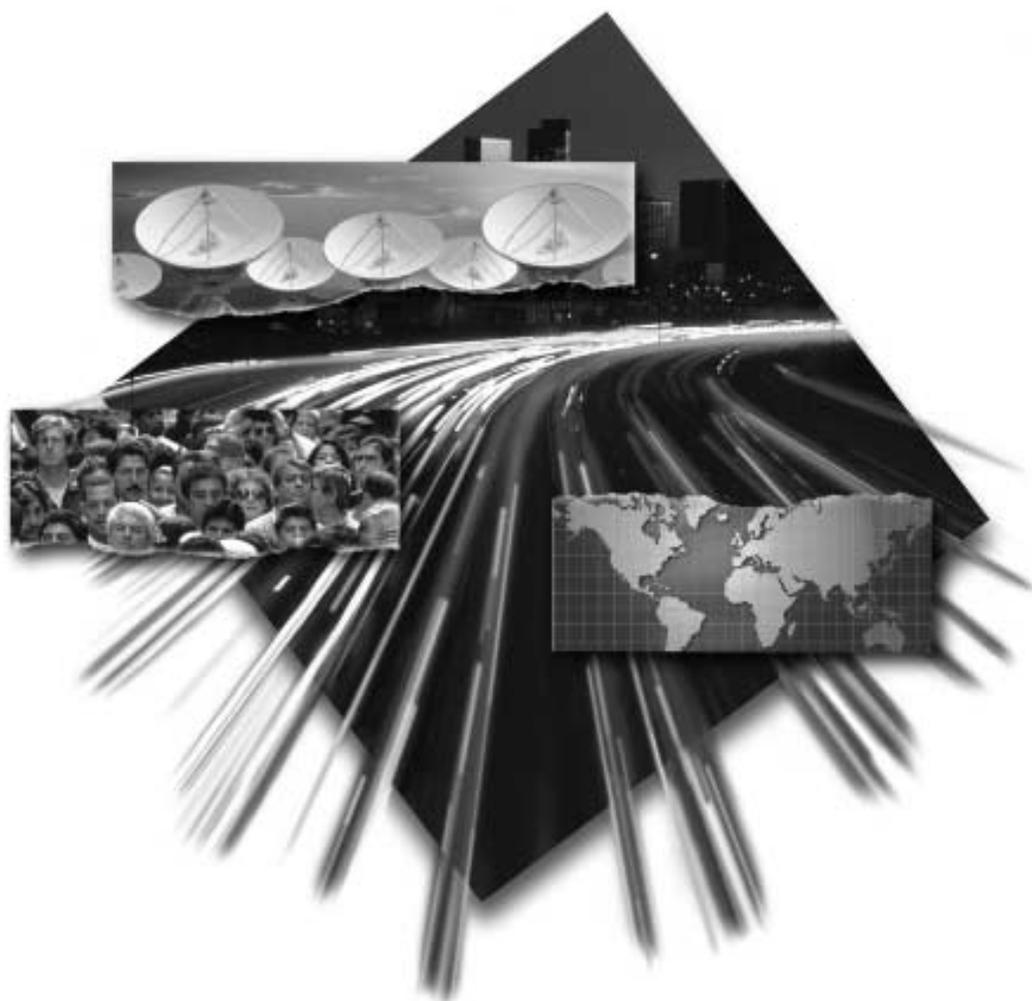
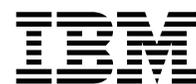


IBM Internet Connection Secure Server

Webmaster's Guide

Version 4.2 for OS/2 Warp





IBM Internet Connection Secure Server

Webmaster's Guide

Version 4.2 for OS/2 Warp

Note

Before using this information and the product it supports, be sure to read the general information under Appendix B, "Notices" on page 303.

First Edition (March 1997)

This edition applies to Version 4.2 of the Internet Connection Secure Server for OS/2 Warp and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

IBM welcomes your comments. A form for readers' comments may be at the back of this publication. If the form has been removed, you may send your comments to the following address:

International Business Machines Corporation
Department CGM
PO BOX 12195
Research Triangle Park, North Carolina
27709-9990
USA

If you prefer to send comments electronically, use one of the following methods:

- IBMLink: CIBMORCF at RALVM13
- IBM Mail: USIB2HPD at IBMMAIL
- Internet: USIB2HPD@VNET.IBM.COM
- Fax: 1-800-227-5088

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1996, 1997. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

| | |
|-------------------------------------|------|
| Welcome! | vii |
| Conventions in this book | vii |
| Information road map | vii |
| What's new in Version 4.2 | viii |

Part 1. Basic Configuration 1

| | |
|--|----|
| Chapter 1. Changing the default configuration | 3 |
| Configuring your server | 4 |
| Controlling access to the Configuration and Administration forms | 6 |
| Changing the default home page | 7 |
| Running your server as a proxy | 11 |
| Setting up your proxy server | 12 |
| Running your server with multiple IP addresses or virtual hosts | 19 |
| Backing up files | 22 |
| | |
| Chapter 2. Starting and stopping your server | 23 |
| Starting the server | 23 |
| Stopping the server | 25 |

Part 2. Advanced Configuration 27

| | |
|--|-----|
| Chapter 3. Using the configuration file | 29 |
| Overview of directives | 36 |
| Basic - Specify required settings | 37 |
| Directories and Welcome Page - Set viewing options | 41 |
| Logging and Reporting - Customize access and error logs and generate access reports | 54 |
| Access control - Set up access control for the server | 76 |
| Security - Set up network security for the server | 89 |
| Multi-format processing - Define file extensions for multi-format processing | 91 |
| Resource mapping - Redirect URLs | 101 |
| Error message customization - Customize error messages the server returns to clients | 111 |
| Timeouts - Close connections automatically | 116 |
| Methods - Set method acceptance | 118 |
| Meta-Information - Name meta-information files and directories | 120 |
| ICAPI application processing - Specify ICAPI applications for request processing | 122 |
| Servlet API Support - Configure the server for Java servlet API support | 130 |
| Proxy server settings - Configure server as a proxy | 134 |
| Performance settings - Define performance settings | 144 |
| System Management - Define system management settings | 150 |

| | |
|---|-----|
| Chapter 4. Customizing logs and reports | 153 |
| Tailoring the logs your server keeps | 153 |
| Tailoring the reports your server creates | 161 |
| Chapter 5. Customizing your Web site | 171 |
| Displaying page count, date, time, and text on a Web page | 171 |
| Using server-side includes to insert information into CGI programs and HTML documents | 178 |
| Chapter 6. Rating Web sites and serving rated Web information | 187 |
| Who can rate Web sites | 187 |
| How Web clients use PICS | 188 |
| How the Internet Connection Secure Server helps you manage PICS labels | 189 |
| How to manage PICS labels from a central file | 192 |
| How to create PICS labels | 195 |
| How to request PICS label information | 195 |
| How to update the PICS configuration file | 196 |
| Chapter 7. Protecting your server | 201 |
| Protection methods | 201 |
| Step 1. Activating protection | 203 |
| Step 2. Passing the requests | 203 |
| Step 3. Deciding what type of protection to use | 204 |
| Step 4. Creating protection setups | 207 |
| Step 5. Limiting access to individual files | 214 |
| Using server group files | 214 |
| Using Access Control List (ACL) files | 215 |
| Protection example (without SSL client authentication) | 218 |
| Protection example (with SSL client authentication) | 221 |
| Chapter 8. Managing your Web server | 223 |
| Simple Network Management Protocol | 223 |
| Monitoring server performance and status | 234 |

Part 3. Security 237

| | |
|--|-----|
| Chapter 9. Making your communications secure | 239 |
| Security concepts | 239 |
| Using the security built into the server | 244 |
| Managing your keys, certificates, and trusted roots | 245 |
| Chapter 10. Using Secure Sockets Layer (SSL) | 255 |
| Overview of SSL security | 255 |
| Step 1. Setting up SSL | 256 |
| Step 2. Specifying SSL client authentication in protection setups and ACL files | 267 |
| Step 3. Using SSL with your server | 267 |
| Chapter 11. Acting as a certification authority for a private Web network | 269 |

| | |
|---|------------|
| Becoming a CA | 269 |
| Processing certificates as a CA | 276 |
| Chapter 12. Supported key lengths and encryption modes | 283 |
| Public and private keys | 283 |
| SSL encryption modes | 284 |

Part 4. Appendixes 285

| | |
|--|------------|
| Appendix A. Command reference | 287 |
| certutil command | 287 |
| cgiparse command | 288 |
| cgiutils command | 291 |
| htadm command | 293 |
| htimage command | 295 |
| httpd command | 298 |
| Appendix B. Notices | 303 |
| Programming interface information | 304 |
| Trademarks | 304 |
| Glossary | 307 |
| Bibliography | 311 |
| For AIX | 311 |
| For HP-UX | 311 |
| For OS/2 Warp | 311 |
| For Solaris | 312 |
| Related publications | 312 |
| Index | 313 |

Welcome!

This book describes the IBM Internet Connection Secure Server for OS/2 Warp.

- Part 1 explains how to change the default home page, how to set up a caching proxy, and how to run your server with multiple IP addresses or virtual hosts.
- Part 2 covers how to modify configuration directives, customize logs and reports on server activity, and protect and manage your server, as well as how to customize your Web site and implement content rating.
- Part 3 describes how to set up security functions.

Conventions in this book

| | |
|-------------------------|--|
| Vertical bar () | When in the left margin, indicates a technical change or addition to the text that was introduced in the English version of this book just prior to General Availability of the Internet Connection Secure Server. In particular, readers of the corresponding non-English books may be interested in checking the areas marked with vertical bars in this book, and also the README file for their product. |
| Boldface | Indicates the name of an item you need to select, the name of a field, or a string you must enter. |
| <i>Italics</i> | Indicates book titles or variable information that must be replaced by an actual value. |
| Monospace | Indicates an example, a portion of a file, or a previously entered value. |

Information road map

Table 1 (Page 1 of 2). Information road map

| If you want to... | Refer to... |
|--|-------------------------|
| Learn about the server | <i>Quick Beginnings</i> |
| Plan for the server | <i>Quick Beginnings</i> |
| Install the server without a response file from a CD-ROM or a remote code server | <i>Quick Beginnings</i> |
| Automatically install the server using a response file | <i>Quick Beginnings</i> |

Table 1 (Page 2 of 2). Information road map

| If you want to... | Refer to... |
|--|---|
| Install the server using SystemView | <i>Quick Beginnings</i> |
| Start and stop the server | Chapter 2, "Starting and stopping your server" on page 23 |
| Use built-in HTML forms to configure the server | Chapter 1, "Changing the default configuration" on page 3 |
| Use the configuration file to configure the server | Chapter 1, "Changing the default configuration" on page 3 |
| Configure a proxy server | Chapter 1, "Changing the default configuration" on page 3 |
| Change the directives in the configuration file | Chapter 3, "Using the configuration file" on page 29 |
| Use the server commands | Appendix A, "Command reference" on page 287 |
| View online documentation | <i>Quick Beginnings</i> |
| Print or order copies of the documentation | <i>Quick Beginnings</i> |
| Protect your server from unwanted access | Chapter 7, "Protecting your server" on page 201 |
| Use SSL V3 functions to make your communications secure | Chapter 9, "Making your communications secure" on page 239 |
| Write programs that can interface with the server | <i>Web Programming Guide</i> |
| Use server-side includes to imbed information in HTML files | "Using server-side includes to insert information into CGI programs and HTML documents" on page 178 |
| Use multiple IP addresses or virtual hosts to manage your server | "Running your server with multiple IP addresses or virtual hosts" on page 19 |
| Customize your error messages | "Error message customization - Customize error messages the server returns to clients" on page 111 |
| Review style guidelines and tips for creating Web information | http://www.ics.raleigh.ibm.com/ |
| Create HTML documents | http://www.ics.raleigh.ibm.com/ |

What's new in Version 4.2

Enhancements for this version include additional function and programming language support, as well as performance improvements.

Customized response based on requesting browser

Your server can now automatically detect the type of browser making a request and return Web information that is appropriately formatted for the particular browser.

Counter and date/time image support for Web pages

You can enhance your Web pages with a user-access counter and the date and time as graphical images.

Enhancements to logging and generating reports on server activity

This version offers a new Java Graphical User Interface (GUI). You can log and analyze activity on your server based on URLs, host names, methods, or return codes. You can generate tailored reports based on associations such as URL hits by host, host hits by URL, return codes by URL, and methods by URL.

HTTP Version 1.1 compliance

Support for HTTP now includes HTTP 1.1 compliance, which provides for persistent connections and virtual hosts. Persistent connections allow the server to accept multiple requests and to send responses over the same TCP/IP connection. Virtual hosts allow one IP address to serve multiple files instead of requiring different IP addresses for different files.

Expanded Common Gateway Interface (CGI) support

CGI support now includes the Java programming language in addition to the other languages already supported, such as C, REXX, and Perl.

Year 2000 support

The Internet Connection Secure Server smooths the transition into the 21st century with reliable support for the year 2000 and beyond.

Performance improvements

Dramatic improvements in single and multiple processor environments coupled with significant CPU requirement reductions provide higher throughput and shorter response time.

Web site content rating support

Based on the Platform for Internet Content Selection (PICS), you can use an independent rating service to rate the content of documents served through your Web site, or you can act as a rating service for your own or other Web sites. Users who request the ratings can then evaluate the Web pages for acceptable content before viewing them.

Client authentication and other enhancements to SSL support

The vast majority of users are choosing the Secure Sockets Layer (SSL) protocol for their secure transactions instead of S-HTTP, so SSL is where we're putting our efforts. We've removed S-HTTP support in this and future versions of the Internet Connection Secure Server, and we will continue to enhance our support of SSL. Our support of SSL V3 now includes client authentication, which is a security enhancement that allows the server to verify, with a high degree of confidence, that any client is who it says it is. Our support of SSL V3 also includes server authentication, data encryption, and additional Message Digest Hashing algorithms.

Simple Network Management Protocol (SNMP) support

The Internet Connection Secure Server now includes an SNMP subagent, which maintains server information and performance data in an SNMP Management Information Base (MIB). From any SNMP-capable network manager, such as IBM NetView for AIX, TME10 Distributed Monitoring, or HP OpenView, you can display, monitor, and adjust thresholds for your server performance.

SOCKS support and SSL tunneling

The Internet Connection Secure Server can now provide proxy support through a SOCKS server. SSL tunneling allows the server to act as a proxy server for secure transactions.

Online access to server performance and status information

Through your browser, you can use the Server Activity Monitor to observe performance and status information about your server and the network. Access log entries are also displayed.

Internet Connection Application Program Interface (ICAPI) support

Improvements include expanded diagnostic capability with additional trace capability and additional messages. Users may use existing HTTP methods or tailor them to their needs.

Java servlet support

Java 1.0 support provides Sun Microsystems, Inc., standards for servlets. Servlets typically offer significant performance enhancements over CGI programs.

Proxy authentication

You can set up password protection for your proxy server independently of the password protection at the destination server.

Book title change

Our *Up and Running* books of previous releases have been renamed *Quick Beginnings*. The focus remains the same: to provide the minimal, essential information for planning for, installing, and starting to use your server as quickly and easily as possible.

New configuration directives and subdirectives:

| | |
|--------------------------|--------------------------------|
| AddCharSet | ReportDataExpire |
| AddClient | ReportDataCompressionProgram |
| AgentLog | ReportDataUnCompressionProgram |
| CGIErrorLog | ReportDataCompressionSuffix |
| CacheLocalMaxBytes | ReportDataSizeLimit |
| CacheLocalMaxFiles | ReportProcessOldLogs |
| DisInheritEnv | Servlet |
| EnableJavaServletSupport | ServletDir |
| LiveLocalCache | ServletLog |
| MaxActiveJavaThreads | SSL_ClientAuth |
| MaxPersistRequest | SNMP |
| PersistTimeout | SNMPCommunityName |
| PICSDBLookup | SocksServer |
| ProxyAccessLog | SSLClientAuth |
| RefererLog | WebMasterEmail |
| ReportDataArchive | |

Deleted configuration directives and subdirectives:

| | |
|-------------------|------------------|
| CacheClean | MinActiveThreads |
| Crypt | PostCrypt |
| DeleteCrypt | POST-Script |
| DELETE-Script | PutCrypt |
| GetCrypt | PUT-Script |
| IdleThreadTimeout | Search |

Changed configuration directives:

| | |
|---------|------------|
| DefProt | NameTrans |
| Disable | Pass |
| Enable | Protect |
| Exec | Redirect |
| Fail | ServerInit |
| Imbeds | ServerTerm |
| Map | Service |
| | Welcome |

Part 1. Basic Configuration

Chapter 1. Changing the default configuration

Your server is operational once you install it. After you have the server installed and running, you will probably want to change some parts of the default configuration file to make the server meet your own particular needs.

This chapter explains what you can change about your server's configuration, as well as how to change these items, by using the Configuration and Administration forms or by editing the configuration file.

| | |
|--|----|
| Configuring your server | 4 |
| Using the Configuration and Administration forms | 4 |
| Editing the configuration file | 5 |
| Controlling access to the Configuration and Administration forms | 6 |
| Changing the default home page | 7 |
| Understanding the document root directory | 8 |
| Changing your document root directory | 8 |
| Understanding the welcome pages | 9 |
| Creating your own home page | 9 |
| Running your server as a proxy | 11 |
| Caching documents on your proxy server | 12 |
| Setting up your proxy server | 12 |
| Step 1. Configure basic proxy functions | 13 |
| Step 2. Configure basic caching functions | 13 |
| Step 3. Designate a port number for your proxy server | 14 |
| Step 4. Configure advanced proxy functions | 14 |
| Step 5. Configure advanced caching functions | 15 |
| Step 6. Specify which clients can use the proxy | 17 |
| Step 7. Set up a secure connection | 18 |
| Running your server with multiple IP addresses or virtual hosts | 19 |
| Multiple IP addresses | 20 |
| Virtual hosts | 20 |
| Setting up your server to use multiple IP addresses or virtual hosts | 20 |
| Backing up files | 22 |

Changing configurations

Configuring your server

You can configure your server either by using the Configuration and Administration forms or by editing the server's configuration file.

Using the Configuration and Administration forms

The server comes with Configuration and Administration forms. These forms are a combination of CGI programs and HTML forms that provide an easy way for you to configure your server or to view your server's current configuration settings.

Once the server is running, you can access the Configuration and Administration forms from WebExplorer (or any other Web browser). The browser can be on the same machine as the server or on any remote client that has access to the server.

To use the Configuration and Administration forms:

- 1 Disable caching on your browser. Also, if you are configuring your server remotely from a browser that uses that specific server as its proxy server, you should disable the proxy server setting on your browser.
- 2 Using your browser, go to the server's Front Page by typing the following URL:
http://your.server.name/
where *your.server.name* is the fully qualified name of your host. For example, <http://www.ibm.com/>
- 3 Click **Configuration and Administration Forms**.
- 4 If you have not used the Configuration and Administration forms since starting your browser, you will be prompted for a user name and password.
Enter the user name and password you specified in the **Administrator ID** and **Password** fields during installation. If you did not change the installation defaults, the authorized user name is `webadmin` and the authorized password is `webibm`. If you did not change the defaults during installation, it is strongly recommended that you do so now to prevent unauthorized access to your server configuration. See "Controlling access to the Configuration and Administration forms" on page 6.
After you enter an authorized user name and password, you go to the Configuration and Administration forms page.
- 5 From the Configuration and Administration forms page, you can link to each of the input forms by clicking on the form name.

Changing configurations

When you go to a form, it is displayed with the current configuration values in its input fields. (If you haven't changed your configuration since installation, these are the default values.)

- 6 From any form, enter information about how you want to configure that particular part of your server.

Each form provides instructions to assist you in deciding what changes to make. For further information, you can click the help icon at the bottom of each form. The help icon links you to a help page that provides detailed steps for using the form to perform particular tasks.

- 7 After you fill in the form, you must click **Apply** to indicate you want to update the server configuration with the changes you made. The **Apply** button is located below the input fields on each form.

If you decide you do not want to use the changes you made to the form, click **Reset**. This returns the fields on the form to the values they had when you first came to the form.

Note: A few of the form pages have more than one set of **Apply** and **Reset** buttons. These pages are actually treated as multiple forms. If you click **Apply** or **Reset**, the action takes place only for the portion of the page associated with that set of buttons.

- 8 If you clicked on **Apply**, the server shows you a message indicating whether your input was accepted. If the input was accepted, you see a Confirmation page that tells you what configuration directives were updated.

If the input was not accepted, you see a Configuration Error page explaining what was wrong with the information you entered. Go back to the form and try to correct the information. From the form you may also want to click the help icon at the bottom of the form.

- 9 If the Confirmation page contains a **Restart Server** button, you can click it to have the server restart and begin using the configuration changes you just made.

If the Confirmation page does not contain a **Restart Server** button, then you need to stop your server and start it again for the configuration changes you made to take effect. For instructions on stopping and starting your server, see Chapter 2, "Starting and stopping your server" on page 23.

Note: From the Confirmation page or the Restart Confirmation page, use the **Configuration Page** button to return to the Configuration and Administration forms.

Editing the configuration file

The other way to configure your server is by editing the configuration file.

By default, the configuration file is named `httpd.conf` and is in the path specified on the `SET ETC` statement in your `CONFIG.SYS` file.

Changing configurations

The configuration file is made up of statements called *directives*. You change your configuration by editing the configuration file, updating the directives, and saving your changes.

When you restart the server, your changes take effect, unless you changed one of the following directives:

- Port
- BindSpecific
- SSLClientAuth
- SSLMode
- SSLPort
- NormalMode
- KeyFile
- imbeds
- ServerRoot

If you changed one of the directives in the above list, you must stop the server and start it again. For instructions on stopping and starting your server, see Chapter 2, “Starting and stopping your server” on page 23.

Chapter 3, “Using the configuration file” on page 29 describes each of the configuration file directives.

Controlling access to the Configuration and Administration forms

After installation, your server has one authorized user name and one password that can be used to access the Configuration and Administration forms.

You specified the authorized user name and password in the **Administrator ID** and **Password** fields during installation. If you did not change the installation defaults, the authorized user name is webadmin and the authorized password is webim.

The user name and password are stored in the ADMIN.PWD password file, which is located in the path specified by the SET ETC statement in your CONFIG.SYS file.

If you have not yet changed the default user name or password, you should do so now to prevent unauthorized access to the Configuration and Administration forms.

You can use either the htadm command or the Configuration and Administration forms to control user names, passwords, and password files.

Changing configurations

If you just want to change the password for the default user name with the `htadm` command, you would enter the following:

```
htadm -passwd d:\setetc\ADMIN.PWD user-name password
```

In the above example:

- `d:\setetc\` is the path specified on the SET ETC statement in your CONFIG.SYS file.
- `user-name` is the name you entered for **Administrator ID** at installation. The installation default is `webadmin`.
- `password` is the new password you want to use.

See “htadm command” on page 293 for complete instructions on how to use the `htadm` command.

For more information about limiting access to the resources on your server, see Chapter 7, “Protecting your server” on page 201.

Changing the default home page

One of the first configuration tasks you will want to perform is changing your server so that it returns your own **home page**. The home page is the document that your server returns when a client sends a request that does not point to a specific directory or file. So when a client sends a URL in the format of:

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

your server responds by sending back your home page. When you first start the server, the server uses the Front Page as your home page and uses the default configuration file.

Configuration settings tell your server where to look for your home page and what its name is. So that you can understand how this works, this section contains background information about the document root directory and the welcome page list. The section then describes an easy way to create your own home page and have the server use it instead of the server's Front Page.

Note: Some Web browsers use the term “home page” to also refer to the first page that the browser loads when it is started. This document uses the term only to refer to the server home page.

Changing configurations

Understanding the document root directory

When the server receives requests that do not point to a specific directory, it tries to serve the request from the **document root directory**. Because you want your home page to be returned for requests that do not specify a directory or file name, you need to have your home page in your document root directory.

The document root directory is the directory you specified as your HTML directory during installation. If you used the installation defaults, the document root directory is C:\WWW\HTML.

Changing your document root directory

At some point you may decide to change the directory your server uses for its document root directory.

For example, you might want to change your document root directory if you are creating a new set of HTML documents for your server to use. While you are creating the new documents, you might want to keep them on a directory not accessible to the server.

Follow these steps to change your document root directory:

- 1** From the Configuration and Administration forms page, click **Request Routing**. (If you're not sure how to access the Forms, see "Using the Configuration and Administration forms" on page 4.)
- 2** On the Request Routing form, select the **Replace** button.
- 3** Change the **Index** field to the number of the Pass rule that has /* as its URL request template.
- 4** Change the **Action** field to **Pass**.
- 5** Enter /* in the **URL request template** field.
- 6** Enter your new document root directory in the **Replacement file path** field.
- 7** Click **Apply**.
- 8** From the Confirmation page, click **Restart Server**.

After restarting, the server begins to use the new document root directory.

So now you know where your server looks for your home page, but how does it know which file to return? This is defined by the list of welcome pages.

Understanding the welcome pages

The server's configuration defines a list of file names that the server can use as **welcome pages**. These are the files the server looks for when it receives URL requests that do not specify a file name.

For requests that do not specify either a directory or file name, you want the server to return your home page. As described in the previous section, the server looks for your home page on the document root directory. It then determines which file contains your home page by matching the list of welcome pages to the files in the directory. The first match it finds is the file it returns as your home page.

The default configuration file defines a list of four welcome pages. The order of the welcome pages is important because the server goes down the list from top to bottom when looking for the file it should return.

The default welcome page file names and their default order are:

1. Welcome.html
2. welcome.html
3. index.html
4. Frntpage.html

The server returns the first file it finds that matches a file name in the list. Until you create a Welcome.html, welcome.html, or index.html file and place the file in the document root directory, the server uses Frntpage.html as your home page.

For example, if you are using the default configuration and your document root directory does not contain a file named Welcome.html, or welcome.html, but does contain files named index.html and Frntpage.html, the index.html file is used as your home page.

For more information on welcome pages and how they are used by directories other than the document root directory, see "Directories and Welcome Page - Set viewing options" on page 41.

The easiest way to have your server start returning your own home page is to create the page in a file named Welcome.html and store the file on your document root directory. The next section explains how to do this.

Creating your own home page

You can use any HTML document for your home page.

If you already have a document you want to use as your home page, just copy it into your document root directory and rename it to Welcome.html, welcome.html, or index.html.

Changing configurations

If you want to create a new home page and have your server start using it, you can use the following procedure:

- 1 Using your browser, go to the server's Front Page by typing the following URL:
`http://your.server.name/`
where *your.server.name* is the fully qualified name of your host. For example, `http://www.ibm.com/`.
- 2 Click **Sample Home Page**.
- 3 From WebExplorer select **Save as** from the **File** pull-down menu on the menu bar. (Most other browsers provide a similar function.)
- 4 Save the file with a name of `Welcome.html` and put it in the document root directory. Your home page must be placed in your root directory since this is the directory your server uses to serve requests that do not point to a specific directory.
- 5 Use your choice of editors to edit the new `Welcome.html` file and turn it into your own home page masterpiece.

Need some help with HTML? Depending upon your current level of expertise with HTML, there are many sources of information available.

- The Internet Connection Family Home Page, which is a link from the server's front page, allows you to download beta code, as well as giveaways, retail products, and accessory products.
- A fun and easy way to learn about HTML and generate ideas for your own home page is to go surfing on the Web. Find some pages you like and use your browser to view the HTML markup that makes them work.
- Many samples are available in BonusPak and BonusPak II.

You can write your HTML documents using any editor capable of producing flat text files. However, if you use a plain text editor you will have to key in each of the HTML tags. There are several editors and related tools designed specifically to help users create HTML documents without having to key in all the surrounding tags.

For example, non-commercial users can get a free copy of an editor called HoTMetaL Free from SoftQuad. Go to the following URL for instructions on how to get HoTMetaL Free:

<http://www.sq.com/products/hotmetal/hm-ftp.htm>

When you have documents ready that you want to show the world, you can start to register your server with various databases. For example, the following URLs take you to sites where you can register your server:

http://www.w3.org/hypertext/DataSources/WWW/Geographical_generation/new-servers.html

Changing configurations

Note: The URL above is shown on two lines only because of its length. When entering the URL on your browser, type one continuous string with no spaces.

<http://www.yahoo.com/>
<http://www.lycos.com/>

Each of the two URLs shown above contains links to a form you can use to register your server.

Running your server as a proxy

Clients connected to a **proxy server** can ask the server to retrieve documents for them from other servers.

You can also set up a **firewall machine**, such as the IBM Internet Connection Secured Network Gateway. A firewall machine is connected to both your internal network, or intranet, and the external Internet. Users of the intranet are inside the firewall. The firewall machine can be set up to prevent external machines from reaching your intranet.

Two types of proxy/firewall configurations are available:

1. A proxy server can link to a firewall machine that has a socks server in it. The proxy server is chained to the socks server in the firewall machine and the firewall machine links to the Internet, as shown in Figure 1. This kind of proxy server, also known as a **socksified proxy**, allows users within your intranet to access the Internet.

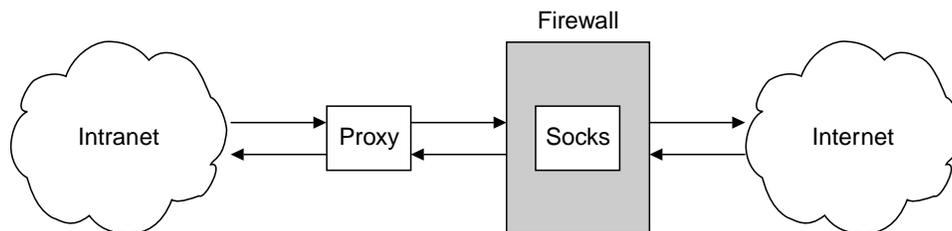


Figure 1. Socksified proxy server and a firewall

2. One machine can contain both the firewall and the proxy server as shown in Figure 2 on page 12. For this kind of configuration, you can use the IBM Internet Connection Secure Server for AIX for your proxy and the IBM Internet Connection Secured Network Gateway for AIX as your firewall.

Changing configurations

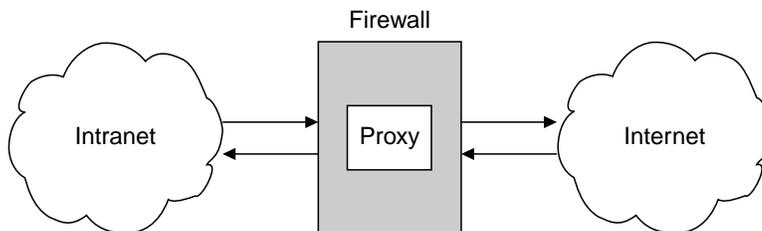


Figure 2. Machine with both a firewall and proxy server

Caching documents on your proxy server

You can also use **caching** to have the proxy server store the documents it retrieves from other servers in a local cache. The server can then respond to subsequent requests for the same documents without having to retrieve them from other servers. This can improve response time.

Within an intranet you may want to set up a server as a caching proxy to reduce the amount of traffic on the network. In large networks you can connect a hierarchy of caching proxies. A client request cascades up through the hierarchy of servers until the document is retrieved from a server's cache or from the actual server where the document resides.

Setting up your proxy server

To set up a proxy, you need to:

- Configure basic proxy functions
- Optionally, configure basic caching functions
- Optionally, designate a port number for your proxy
- Optionally, configure advanced proxy functions
- Optionally, configure advanced caching functions
- Specify which clients can use the proxy
- Set up a secure connection

After you've configured your proxy server, tell your users so they can configure their browsers to point to it.

Step 1. Configure basic proxy functions

To configure your server as a proxy, specify which protocols you want the server to act as a proxy for. You do this by one of two methods:

- Filling in the Proxy Server Settings form for the protocols this server will function as a proxy for
- Adding Pass rules for the protocols to your server's list of mapping rules

Either method enables your server to act as a proxy.

To fill in the form:

- 1** From the Configuration and Administration forms page, under the heading "Proxy Settings," click **Proxy Server Settings** to display the Proxy Server Settings form.
- 2** Select the protocols that the server will act as a proxy for. (Checking a selection box generates a Pass rule for the protocol checked.)
- 3** You should accept the default size specified for the Proxy buffer size.
- 4** If you want your server to be a socksified proxy server, you need to specify the host name or IP address of the socks server through which this proxy server will be passing requests.

When you specify a socks server on this form, you indicate that you want this proxy server to be chained to a socks server.

- 5** Click **Apply**.

Once the changes take effect, your server runs as a proxy.

Step 2. Configure basic caching functions

Perform this step if you want to make your proxy server a caching proxy server.

- 1** From the Configuration and Administration forms page, click **Caching Settings**.
- 2** From the Caching Settings form, check the **Enable proxy caching** box.
- 3** If you have more than 5 megabytes of disk space to use for caching files, change the **Cache size** field to a larger number.
- 4** In the **Root directory for cached files** field, enter the absolute path name of the directory where you want to keep cached files. Make sure the directory you specify has at least as much disk space available as you enter for **Cache size**.
- 5** In the **File name in which to log cache access** field, enter the absolute path and file name of the file you want to use for logging cache access.

Changing configurations

- 6 For the upper size limits for files to be cached, the value you specify indicates that any file bigger than that value is not cached.
- 7 Specify a size for the lower file limit. This limit indicates the minimum file size a file has to be before file size is taken into account when doing garbage collection. The more the size of the file is above this limit, the more likely it is to be removed during garbage collection.
- 8 Click **Apply**.
- 9 Restart your server so your changes will take effect. See Chapter 2, “Starting and stopping your server” on page 23 for more information on starting and restarting.

Step 3. Designate a port number for your proxy server

Perform this step only if you want your proxy server to listen to a port number other than the HTTP default of 80.

- 1 From the Configuration and Administration forms page, click **Basic**.
- 2 Change the **Default port number** field to the port you want the proxy to listen to. The default value for this field is 80. Some other commonly used port numbers for proxy servers are 8080 and 8008.
- 3 Click **Apply**.
- 4 Restart your server so your changes will take effect. See Chapter 2, “Starting and stopping your server” on page 23 for more information on starting and restarting.

Step 4. Configure advanced proxy functions

You can specify that certain protocol requests are routed to another proxy server. This allows you to chain together a hierarchy of proxy servers.

- 1 From the Configuration and Administration forms page, click **Proxy Chaining and Non-Proxy Domains**.
- 2 In the Proxy chaining section, select the **Insert before** button.
- 3 From the selection list, select the protocol you are specifying a URL for.
- 4 In the URL field, type the URL where you want requests for that protocol to be directed.
- 5 Click **Apply**.

Changing configurations

If you've set up a hierarchy of proxies, you can specify any domain names the proxy can get to directly rather than going to the next proxy in the hierarchy. These are called non-proxy domains.

- 1 From the Configuration and Administration forms page, click **Proxy Chaining and Non-Proxy Domains**.
- 2 In the Non-proxy domains section in the non-proxy domains field, type the name of the domain or domains to which requests should be passed without going through the next proxy in the hierarchy.
- 3 Click **Apply**.

Step 5. Configure advanced caching functions

Several other functions are available to help you control the caching functions. You can control:

- How long HTTP and FTP files are cached
- How often the cache is cleaned
- How expired files are removed from the cache
- Which files you want your server to cache

Controlling how long HTTP files are cached

To control how long HTTP or HTTPS files are kept in the cache after they are last used, do the following:

- 1 From the Configuration and Administration forms page, click **Time Limit for Unused Cached HTTP Files**.
- 2 Specify the amount of time in months, days, weeks, or hours that you want the server to keep unused cached HTTP files.
- 3 Specify the amount of time in months, days, weeks, or hours that you want the server to keep unused cached HTTPS (SSL) files.
- 4 Click **Apply**.

Controlling how often the cache is cleaned

To control how long FTP files are kept in the cache and how long to keep unused FTP files in the cache, do the following:

- 1 From the Configuration and Administration forms page, click **Expiration Settings for Cached FTP Files**.

Changing configurations

- 2 For the default expiration time, specify the amount of time (in months, days, weeks, or hours) to keep FTP files in the cache, regardless of whether they are used.
- 3 For the unused expiration time, specify the amount of time (in months, days, weeks, or hours) to keep unused FTP files in the cache. In general, unused expiration time should be lower than default expiration time.
- 4 Click **Apply**.

Controlling how expired files are removed from the cache

To specify how you want expired files removed from the cache, do the following:

- 1 From the Configuration and Administration forms page, click **Cache Storage Reuse**.
- 2 The box to enable storage reclamation is checked. Delete the check, if you do not want to enable garbage collection.
- 3 Specify the maximum memory (in kilobytes) available for garbage collection. The process works best if it can read all cache information into memory at one time. The amount of memory needed will vary based on dynamic changes such as the directory structure of cached files. If garbage collection fails because there is not enough memory on your system, set this field to a smaller value. If you have plenty of memory to spare, you may want to set this value above the default of 500.
- 4 Specify the time you want garbage collection to occur. Select a time when the server won't be busy. Garbage collection consumes system resources, and the cache is disabled when garbage collection is being done.
- 5 Click **Apply**.

Controlling which files you want your server to cache

To specify which files you want your server to consider for caching, you can create a list of URLs you do want to cache or a list of URLs you do want not to cache. If you do not create either of these lists, any file is a candidate for caching. The server considers files for caching if the request is a GET, the URL does not contain a question mark (?), and the requested file is not protected by a user ID/password.

- 1 From the Configuration and Administration forms page, click **Caching Filters**.
- 2 Click **Add**.

Changing configurations

- 3 Click either **Cache the URL requests listed below** or **Do not cache the URL requests listed below**.
- 4 Type in the list of URLs.
- 5 Click **Apply**.

Step 6. Specify which clients can use the proxy

You can use the server's protection function to control which clients can use your server as a proxy.

The default configuration file contains commented lines that you can use as a basis for controlling access to your proxy. For this reason, it is easier to accomplish this task by editing the configuration file than by using the Configuration and Administration forms.

Follow these steps to define which clients can use your server as a proxy:

- 1 Use the text editor of your choice to open the configuration file.

By default, the configuration file is named `httpd.conf` and is in the path specified on the `SET ETC` statement in your `CONFIG.SYS` file.

- 2 Find the following commented Protection and Protect directives:

```
# Protection PROXY-PROT {
#     ServerId      YourProxyName
#     Mask          @(*.ibm.com, 128.141.*.*, *.ncsa.uiuc.edu)
# }
# Protect http:*    PROXY-PROT
# Protect ftp:*     PROXY-PROT
# Protect gopher:*  PROXY-PROT
# Protect https:*   PROXY-PROT
# Protect wais:*    PROXY-PROT
# Protect news:*    PROXY-PROT
```

The Protect statement for `https` is in the configuration file.

- 3 Remove the comment character (`#`) from each of the lines.
- 4 Change the Mask subdirective so that it contains templates for the host names or IP addresses of the clients that need to use your server as a proxy.

In order to use host name templates, you must set the `DNS-Lookup` directive to `On`. If the `DNS-Lookup` directive is set to `Off` (the default), you can use IP address templates only. See "DNS-Lookup - Specify whether you want to look up host names of clients" on page 38.

The Internet Connection Secure Server distinguishes between a user ID and password used for authentication at the proxy server and those used for authentication at the end-point server.

Changing configurations

When a user ID is specified in the protection mask for a proxy request, the proxy server requires the browser to send a PROXY-AUTHENTICATE header. If the header is missing or incorrect, the proxy server sends a message (HTTP 407 error status) back to the browser stating that proxy authentication is required.

If request authentication is required at the end-point server, that server sends back a message (HTTP 401 error status) stating that authentication is required. The end-point server's error message is returned to the browser by the proxy server. A WWW-AUTHENTICATE header sent by the browser is forwarded by the proxy server to the end-point server.

PROXY-AUTHENTICATE headers are never forwarded by a proxy server; this ensures that validated proxy user IDs and passwords are not sent to servers that they are not intended for.

Note: Web browsers must also support proxy authentication if you require user ID authentication in your proxy protection setup mask.

- 5 Save the configuration file with the changes you make.
- 6 Restart the server.

The server will now act as a proxy only for clients and requests that meet the specifications on the mask subdirectives.

Step 7. Set up a secure connection

If you are using SSL, the kind of connection between the client and the proxy and between the proxy and the destination server can vary depending on whether the client uses SSL tunneling.

Connections with SSL tunneling

Some clients, such as Netscape Navigator, use SSL tunneling to establish a secure connection to a destination server through a proxy. The proxy can be a base or secure server.

Because SSL tunneling is generic, it can be used to access resources on different ports. For example, the Internet Connection Secure Server uses port 443 for **https** requests and Netscape uses port 563 for SNEWS (Secure News).

To set up a proxy for SSL tunneling:

- 1 Enable the CONNECT method. You can use the Method Enablement form or code the following directive in the proxy's configuration file:

```
Enable CONNECT
```
- 2 Specify a Pass rule for each port connected to a destination server. You can use the Request Routing form or code the following in the proxy's configuration file.

Changing configurations

Examples:

For **https** requests, specify Pass *:443

For SNEWS (Secure News), specify Pass *:563

- 3 Have clients set their security proxy in Netscape Navigator to point to port 80.

Connections without SSL tunneling

Secure WebExplorer does not use SSL tunneling. You need Pass directives defined for **https** and **http** requests.

Here's how the connection from a Secure WebExplorer client to a destination server through a proxy is handled:

- The kind of connection between the client and the proxy is governed by the proxy protocol. If the proxy protocol is **https**, then the connection between the client and the proxy will be secure. If the proxy protocol is **http**, then the connection between the client and the proxy will not be secure.
- The kind of connection from the proxy to the server is governed by what is specified in the URL. If the URL specifies **https**, the connection between the proxy and the server will be secure. If the URL specifies **http**, the connection between the proxy and the server will not be secure.
- The lock will be displayed on Secure WebExplorer if the URL specifies **https**. This means, at minimum, there is a secure network connection between the proxy and the server. There will also be a secure network connection between the client and the proxy if the proxy protocol is **https**.

The server's certificate will not be displayed on Secure WebExplorer if the URL specifies **https**.

Running your server with multiple IP addresses or virtual hosts

You may want to use one server to provide Web sites for multiple customers. For example, you might have two customers (customerA and customerB), both of whom want to make information about their companies available on the World Wide Web. You might want to put both Web sites on the same machine if the expected number of requests for the information is not great enough to justify a separate machine for each customer.

With the Internet Connection Secure Server, you can use multiple IP addresses, virtual hosts, or both to provide multiple Web sites on one server.

Changing configurations

Multiple IP addresses

To use multiple IP addresses, your server must be installed on a machine with multiple network connections.

If you have only one network connection and run two instances of the server on the same machine, then only one server has the benefit of using the default port number. Requests to the other server would have to include a port number.

If your 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 you would define a different host name. So customerA could be www.customerA.com on IP address 9.67.106.79 and customerB could be www.customerB.org on IP address 9.83.100.45. You could 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 would not require a port number.

Virtual hosts

With virtual hosts, no additional hardware is required, and you can save IP addresses. However, clients must support HTTP 1.1 or HTTP 1.0 with 1.1 Extensions.

With virtual hosts, you can run just one instance of the server and assign each customer to a different host. In the domain name server, you define your hosts and associate them with the IP address of your server. You can then configure the server to serve a different set of information depending on the host for which the request is made. Requests do not require a port number.

Setting up your server to use multiple IP addresses or virtual hosts

Setting up your server to use multiple IP addresses or virtual hosts is very similar. For multiple IP addresses, you'll need to specify the IP address a request comes in on and for virtual hosts, you'll need to specify the host name for which a request is made.

You configure the server to serve different information for each customer by indicating that certain parts of your configuration apply only to requests coming in on certain addresses or for certain hosts. You can configure three key parts of your server so that requests are processed differently based on the IP address they come in on or the host name in the URL.

- Welcome pages
- Mapping rules
- Access control

Changing configurations

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 or the host name in the URL. 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` or for `hostA`, and `homeB.html` is a welcome page only for requests received on an address `9.83.100.45` or for `hostB`.

From the Configuration and Administration forms page, you can configure your 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 a host name.

Alternatively, if you are editing the configuration file, you can add an *IP-address* or *hostname* at the end of a `Welcome` directive to associate a welcome page file name with an IP address or a host name. For details, see the description of the `Welcome` directive in “Directories and Welcome Page - Set viewing options” on page 41.

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 or the host name in a URL. 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 `/cgi-bin/` received on address `9.67.106.79` or for `hostA` is mapped to the `/customerA/cgi/` directory, and the same request received on `9.83.100.45` or for `hostB` is mapped to the `/customerB/cgi/` directory.

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

Alternatively, if you are editing the configuration file, you can add an *IP-address* or *hostname* at the end of `Exec`, `Fail`, `Map`, `Pass`, and `Redirect` directives to associate the directive with an IP address or a host name. For details, see the description of these directives in “Resource mapping - Redirect URLs” on page 101.

Access Control

You can activate different protection rules for a request based on the address the request comes in on or the host name in a URL. Protection rules are defined in protection setups and determine how your server controls access to files and programs.

For example, you might want to specify that a request beginning `/cgi-bin/` received on address `9.67.106.79` or for `hostA` is protected by the rules in a protection setup named `PROT-A` and the same request received on `9.83.100.45` or for `hostB` is protected by the rules in a protection setup named `PROT-B`.

From the Configuration and Administration forms page, you can configure how protection is activated by clicking on **Document Protection**. From the Document Protection form, click the help icon for information on protecting documents and how to associate protection with an IP address or a host name.

Alternatively, if you are editing the configuration file, you can add an *IP-address* or *hostname* at the end of `DefProt` and `Protect` directives to associate the directive with an IP address or a host name. For details, see the description of these directives in “Access control - Set up access control for the server” on page 76.

Backing up files

It is recommended that you periodically back up the following files:

- Configuration files (`httpd.conf`, `ics_pics.conf`)
- Password files
- Group files
- ACL files
- Content files

For a secure server, you should also back up:

- Key ring files
- Signed certificates and, optionally, certificate requests

Chapter 2. Starting and stopping your server

This chapter contains basic instructions for starting and stopping your server. It also describes the first tasks you will probably want to perform after installation.

| | |
|--|----|
| Starting the server | 23 |
| Starting automatically from the OS/2 startup folder | 23 |
| Starting from the Internet Connection Secure Server icon | 24 |
| Starting from the command prompt | 24 |
| Starting multiple instances of the server | 24 |
| Restarting from the Internet Connection Secure Server Window | 25 |
| Restarting from the Configuration and Administration Forms | 25 |
| Stopping the server | 25 |

Starting the server

This section describes how to start and stop your server after you have finished installing it. If you think the server is running but you do not see the window for it on your OS/2 Desktop, press **Ctrl+Esc** to see the list of active OS/2 programs. If the Window List contains an entry for the server, double click the entry to bring the server's window into view.

If you are running a non-secure server, you will see the message Security not available. *filename.dll* did not load. This message lets you know that you are not running the Internet Connection Secure Server. You do not need to do anything in response to the message.

Starting automatically from the OS/2 startup folder

If during installation you checked Auto Start Server at Bootup, the server has been added to your OS/2 Startup folder. Each time you start or reboot your system, the server will start. If you rebooted your system after installing the server, the server should be running now.

When you start the server automatically, it uses the current configuration settings. If you have not changed the server configuration since installation, the current

Starting and Stopping Your Server

configuration is based on the information you entered during installation along with defaults.

Starting from the Internet Connection Secure Server icon

The installation procedure creates an Internet Connection Secure Server folder on your OS/2 Desktop. Inside the folder is an icon for the Internet Connection Secure Server. To start the server, double click the icon.

The server starts with the current configuration settings. If you have not changed the server configuration since installation, the current configuration is based on the information you entered during installation along with defaults.

Starting from the command prompt

You can start the server from any OS/2 command prompt by entering:

```
httpd
```

This starts the server with the current configuration settings. If you have not changed the server configuration since installation, the current configuration is based on the information you entered during installation along with defaults.

The httpd command also accepts various options for changing configuration settings at run time. For details on the httpd command options, see "httpd command" on page 298.

Starting multiple instances of the server

You can start multiple instances of the server, but each instance must listen on a separate port. All instances, other than the first, can be started by entering the following command from a command prompt:

```
httpd -r other_rule_file
```

where *other_rule_file* is a configuration file that specifies the individual port.

Restarting from the Internet Connection Secure Server Window

You can restart your server from the Internet Connection Secure Server window by selecting **Restart** from the **Server** pull-down menu on the menu bar. The server reloads the configuration settings from the configuration file it was originally started with.

Most changes you have made to the configuration file will be used by the server after it restarts. However, if you changed a port setting or security settings, you need to stop your server and start it again for those changes to take effect.

Restarting from the Configuration and Administration Forms

When using the Configuration and Administration Forms, you are sometimes given the option of restarting the server after you apply changes. See "Using the Configuration and Administration forms" on page 4 for more information.

Stopping the server

You can stop the server the same ways you stop other OS/2 programs. Do any of the following:

- Go to the Internet Connection Secure Server window and double click the small icon in the top left corner.
- Go to the Internet Connection Secure Server window and select **Exit** from the **Server** pull-down menu on the menu bar.
- Go to the Internet Connection Secure Server window and press **F3**.
- Press **Ctrl+Esc**. With the right mouse button click the Window List entry for the Internet Connection Secure Server. Click **Close**.

Part 2. Advanced Configuration

Chapter 3. Using the configuration file

This chapter describes each server configuration directive. If you choose to configure the server by editing the httpd.conf configuration file, use this chapter as a reference. If you choose to configure your server by using the Configuration and Administration forms, you do not need to refer to this chapter.

The directive descriptions are grouped according to function, similar to the way the Configuration and Administration forms are grouped. Within each group, the directives are in alphabetical order.

| Type | Description | Page | Name |
|-------------------------------------|--|------|-----------------|
| Basic | Specify if the server binds to one or all IP addresses | 37 | BindSpecific |
| | Specify whether you want to look up host names of clients | 38 | DNS-Lookup |
| | Specify the fully qualified domain name or IP address for the server | 38 | HostName |
| | Specify whether server-side includes will be dynamically imbedded | 39 | imbeds |
| | Specify the port on which you want the server to listen for requests | 40 | Port |
| | Specify the directory where the server program is installed | 41 | ServerRoot |
| Directories and Welcome Page | Specify the icon URL used to align the heading of directory listings | 41 | AddBlankIcon |
| | Specify the icon URL for directories on directory listings | 42 | AddDirIcon |
| | Bind an icon to a MIME content-type or encoding-type | 43 | AddIcon |
| | Specify the icon URL for a parent directory on directory listings | 43 | AddParentIcon |
| | Specify the icon URL for unknown file types on directory listings | 44 | AddUnknownIcon |
| | Specify if a welcome file is returned for all directory requests | 45 | AlwaysWelcome |
| | Control directory listings | 45 | DirAccess |
| | Control directory README files | 46 | DirReadme |
| | Use brackets around alternative text on directory listings | 46 | DirShowBrackets |
| | Show byte count for small files on directory listings | 47 | DirShowBytes |

| Type | Description | Page | Name |
|------------------------------|--|-------------|----------------------------|
| | Use case when sorting files on directory listings | 47 | DirShowCase |
| | Show date last modified on directory listings | 47 | DirShowDate |
| | Show descriptions for files on directory listings | 48 | DirShowDescription |
| | Show hidden files on directory listings | 48 | DirShowHidden |
| | Show icons in directory listings | 48 | DirShowIcons |
| | Set the maximum description length on directory listings | 49 | DirShowMaxDescrLength |
| | Set the maximum length for file names on directory listings | 49 | DirShowMaxLength |
| | Set the minimum length for file names on directory listings | 49 | DirShowMinLength |
| | Show file size on directory listings | 50 | DirShowSize |
| | Specify the path for the directory listing internal icons | 50 | IconPath |
| | Specify names of welcome files | 50 | Welcome |
| User directories | Specify the directory that contains user subdirectories | 52 | HomeDir |
| | Specify the name of the accessible subdirectory off of user subdirectories | 53 | UserDir |
| Logging and Reporting | Name the path for the access log file | 54 | AccessLog |
| | Remove existing access log files or run a user exit | 55 | AccessLogArchive |
| | Suppress log entries for specific files or directories | 56 | AccessLogExcludeURL |
| | Suppress log entries for files or directories requested by a given method | 57 | AccessLogExcludeMethod |
| | Suppress log entries for specific MIME types | 57 | AccessLogExcludeMimeType |
| | Suppress log entries for specific return codes | 58 | AccessLogExcludeReturnCode |
| | Remove existing access log files when they reach a given age in days | 58 | AccessLogExpire |
| | Remove existing access log files when they reach a given collective size | 59 | AccessLogSizeLimit |
| | Give a short description of the report to be created | 59 | AccessReportDescription |

| Type | Description | Page | Name |
|------|---|------|-------------------------------|
| | Suppress from the report the log entries for specific files or directories | 60 | AccessReportExcludeURL |
| | Include in the report only log entries for specific files directories | 60 | AccessReportIncludeURL |
| | Suppress from the report the log entries for specific host names | 61 | AccessReportExcludeHostName |
| | Include in the report only log entries for specific host names | 61 | AccessReportIncludeHostName |
| | Suppress from the report the log entries of a given method type | 62 | AccessReportExcludeMethod |
| | Suppress from the report the log entries with a given range of return codes | 62 | AccessReportExcludeReturnCode |
| | Name the path for the root directory where access log reports are stored | 63 | AccessReportRoot |
| | Name the report template | 63 | AccessReportTemplate |
| | Specify the top number of items on which to report | 64 | AccessReportTopList |
| | Specify the path for the agent log files | 64 | AgentLog |
| | Specify the path for the cache access log files | 65 | CacheAccessLog |
| | Specify the path for the CGI error log files | 66 | CgiErrorLog |
| | Name the file where you want to log internal server errors | 66 | ErrorLog |
| | Remove existing error log files or run a user exit | 67 | ErrorLogArchive |
| | Remove existing error log files when they reach a given age in days | 68 | ErrorLogExpire |
| | Remove existing error log files when they reach a given collective size | 69 | ErrorLogSizeLimit |
| | Specify common or old log file format | 69 | LogFormat |
| | Specify GMT or local time stamps in log files | 70 | LogTime |
| | Specify whether access log can be seen from graphic user interface | 70 | LogToGUI |
| | Suppress log entries for specific hosts or domains matching a template | 71 | NoLog |
| | Specify the path for the referer access log files | 71 | RefererLog |
| | Specify path to the compression program used for the access data files | 72 | ReportDataCompressionProgram |

| Type | Description | Page | Name |
|--------------------------------|--|----------------|---|
| | Specify path to the uncompression program used for the access data files | 72 | ReportDataUnCompressionProgram |
| | Specify the suffix appended to the compressed file | 73 | ReportDataCompressionSuffix |
| | Specify how to handle old log files not originally included in a report. | 73 | ReportProcessOldLogs |
| | Remove existing access data files when they reach a given collective size. These access data files are named <i>access.mmdyyyy</i> and are used for reports. | 74 | ReportDataSizeLimit |
| | Specify whether to remove existing access data files. These access data files are named <i>access.mmdyyyy</i> and are used for reports. | 74 | ReportDataArchive |
| | Remove existing access data files when they reach a given age. These access data files are named <i>access.mmdyyyy</i> and are used for reports. | 76 | ReportDataExpire |
| Access control | Specify default protection setup for requests that match a template | 77 | DefProt |
| | Activate protection setup for requests that match a template | 79 | Protect |
| | Define a named protection setup within the configuration file | 83 84 87 | Protection Protection subdirectives SSL client authentication subdirectives |
| Security | Set name for key ring file | 89 | KeyFile |
| | Turn port on or off for HTTP | 89 | NormalMode |
| | Turn SSL client authentication on or off | 90 | SSLClientAuth |
| | Turn port on or off for SSL | 90 | SSLMode |
| | Set port for SSL security | 91 | SSLPort |
| Multi-format processing | Specify the language of files with particular extensions | 93 | AddLanguage |
| | Specify the MIME content encoding of files with particular extensions | 94 | AddEncoding |
| | Specify, for a given extensions, the codepage in which the character sets documents are stored | 94 | AddCharSet |
| | Specify the data type of files with particular extensions | 95 | AddType |
| | Specify whether extension definitions are case sensitive | 99 | SuffixCaseSense |

| Type | Description | Page | Name |
|-------------------------------------|---|------|----------------|
| | Specify file extensions to use when responding to particular browsers | 99 | AddClient |
| Resource Mapping | Run a CGI program for matching requests | 101 | Exec |
| | Reject matching requests | 103 | Fail |
| | Change matching requests to a new result string | 104 | Map |
| | Accept matching requests | 106 | Pass |
| | Send matching requests to another server | 108 | Redirect |
| | Specify which environment variables are inherited by CGI programs | 109 | InheritEnv |
| | Specify which environment variables are disinherited by CGI programs | 110 | DisInheritEnv |
| Error message customization | Specify a customized message for a particular error condition | 111 | ErrorPage |
| Timeouts | Specify input timeout setting | 116 | InputTimeout |
| | Specify output timeout setting | 117 | OutputTimeout |
| | Specify script timeout setting | 117 | ScriptTimeout |
| Methods | Disable HTTP methods | 119 | Disable |
| | Enable HTTP methods | 119 | Enable |
| Meta-Information | Specify name of subdirectory for meta-information files | 121 | MetaDir |
| | Specify the extension for meta-information files | 121 | MetaSuffix |
| ICAPI application processing | Customize the server initialization step | 123 | ServerInit |
| | Customize the user pre-exit step | 123 | PreExit |
| | Customize the authentication step | 124 | Authentication |
| | Customize the name translation step | 124 | NameTrans |
| | Customize the authorization step | 125 | Authorization |
| | Customize the object type step | 126 | ObjectType |
| | Customize the service step | 126 | Service |
| | Customize the PICS label retrieval step | 127 | PICSDBLookup |
| | Customize the data filter step | 128 | DataFilter |
| | Customize the log step | 128 | Log |
| | Customize the error step | 129 | Error |
| | Customize the post-exit step | 129 | PostExit |

| Type | Description | Page | Name |
|----------------------------------|---|------|--------------------------|
| | Customize the server termination step | 130 | ServerTerm |
| Servlet API configuration | Enable the server to support Java servlets | 131 | EnableJavaServletSupport |
| | Specify threads for request processing | 132 | MaxActiveJavaThreads |
| | Specify a log for Java servlet messages | 132 | ServletLog |
| | Specify the directory for Java servlets | 133 | ServletDir |
| | Specify a servlet's initialization parameters | 133 | Servlet |
| Proxy server settings | Specify default expiration time for files that do not have an expiration date | 134 | CacheDefaultExpiry |
| | Turn cache expirations off | 134 | CacheExpiryCheck |
| | Specify fraction of Last-Modified time to be used for determining expiration date | 135 | CacheLastModifiedFactor |
| | Specify lower limit for cached file size | 135 | CacheLimit_1 |
| | Specify upper limit for cached file size | 136 | CacheLimit_2 |
| | Specify how long a file being cached can remain locked | 136 | CacheLockTimeOut |
| | Specify stand alone cache mode | 136 | CacheNoConnect |
| | Cache only files with URLs that match a template | 137 | CacheOnly |
| | Specify cache root directory | 137 | CacheRoot |
| | Specify cache size | 138 | CacheSize |
| | Specify how long to keep unused cached files that match a template | 138 | CacheUnused |
| | Turn proxy caching on/off | 138 | Caching |
| | Specify a proxy server for this proxy to connect to for FTP requests | 139 | ftp_proxy |
| | Turn garbage collection on or off | 139 | Gc |
| | Specify a daily time for garbage collection | 139 | GcDailyGc |
| | Specify how much memory to use for garbage collection | 140 | GcMemUsage |
| | Specify a proxy server for this proxy to connect to for Gopher requests | 140 | gopher_proxy |
| | Specify a proxy server for this proxy to connect to for HTTP requests | 141 | http_proxy |
| | Set the size of the buffer for dynamic data generated by the server. | 141 | MaxContentLengthBuffer |

| Type | Description | Page | Name |
|-----------------------------|--|------|--------------------|
| | Connect directly to domains matching templates | 141 | no_proxy |
| | Do not cache files with URLs that match a template | 142 | NoCaching |
| | Name the path for the proxy access log file | 142 | ProxyAccessLog |
| | Specify the socks server the proxy is chained to | 143 | SocksServer |
| | Specify a proxy server for this proxy to connect to for WAIS requests | 143 | wais_proxy |
| Performance settings | Specify files you want to load in memory at start up | 145 | CacheLocalFile |
| | Specify the maximum amount of memory allocated for file caching | 145 | CacheLocalMaxBytes |
| | Specify the maximum number of files to be simultaneously cached | 146 | CacheLocalMaxFiles |
| | Specify whether the cache is updated when a cached file is modified | 146 | LiveLocalCache |
| | Specify the maximum number of threads to have active | 147 | MaxActiveThreads |
| | Specify the maximum number of requests the server receives on a persistent connection | 148 | MaxPersistRequest |
| | Specify the amount of time the server waits before cancelling a persistent connection | 148 | PersistTimeout |
| | Specify the priority you want your server to have on your system | 147 | ServerPriority |
| | Specify whether meta files will be checked | 149 | UseACLs |
| | Specify whether meta files will be used | 149 | UseMetaFiles |
| System Management | Enable and disable SNMP support | 150 | SNMP |
| | Provide a security password to control access to a particular community of monitored servers | 150 | SNMPCommunityName |
| | Create an e-mail address to receive SNMP problem reports | 151 | WebMasterEmail |

Overview

Overview of directives

Each directive description includes:

- Heading with the directive name and a brief description
- Usage instructions
- Example of how the directive might appear in the configuration file

Each configuration directive follows the same general syntax:

```
DirectiveName value
```

- Default value or values of the directive

These are the original values coded in the default configuration file. (Change only the parts of the configuration file that you want to be different from the default settings.)

For a list of the new directives added in Version 4.2, see page xi.

Some directives (the *value* portion) contain templates for requests, path names, or host names. Except where otherwise indicated, you can use the asterisk (*) character in templates. For the template to be matched, an asterisk can be replaced by any character string or single character.

The examples and defaults in this chapter use the backslash (\) character when showing a path name. In the configuration file, you can actually use either the backslash or forward slash (/) character when specifying a path name. Always use the forward slash (/) when specifying a URL or request template.

Several configuration directives allow you to specify a positive string. You can enter any of the following words:

- Yes
- On
- OK
- Enable

Several configuration directives allow you to specify a negative string. You can enter any of the following words:

- No
- Off
- None
- Disable

Several configuration directives allow you to specify an amount of time. You can specify any combination of:

hh hours

| | |
|------------------|-----------------------------|
| hh:mm | hours and minutes |
| hh:mm:ss | hours, minutes, and seconds |
| n years | number of 365-day years |
| n months | number of 30-day months |
| n weeks | number of 7-day weeks |
| n days | number of 24-hour days |
| n hours | number of 60-minute hours |
| n minutes | number of 60-second minutes |
| n seconds | number of seconds |

All of your entries will be converted to seconds and added together.

The server currently does not allow the use of blank characters within a filename specified in the configuration file. Blank spaces are treated as delimiters.

Basic - Specify required settings

Use the directives described in this section to control your server's basic configuration settings.

BindSpecific - Specify if the server binds to one or all IP addresses

Use this directive on a multi-networking system to run a different server on each IP address. All the servers may listen on the same port.

If you specify On, the server binds to the IP address specified in the HostName directive only, instead of binding to all local IP addresses.

If this directive has not been specified, the server binds to the default HostName.

If you change this directive, you must stop the server and then start it again. The server will not pick up the change if you only restart it.

Example

```
BindSpecific On
```

Initial configuration file setting: None.

Program default setting

```
BindSpecific Off
```

Basic

DNS-Lookup - Specify whether you want to look up host names of clients

Use this directive to specify whether you want your server to look up the host name of requesting clients.

The value you use affects the following things about how your server works:

- The performance of the server. Using the default value of `Off` improves the performance and response time of the server because it does not use resources to perform the host name lookup.
- The information your server records about clients when writing to log files.
 - `Off` - Clients identified by IP address
 - `On` - Clients identified by host name
- Whether you can use host names on address templates in protection setups, server group files, and ACL files.
 - `Off` - Cannot use host names on address templates; must use IP addresses
 - `On` - Can use host names on address templates; cannot use IP addresses

Example

```
DNS-Lookup On
```

Initial configuration file setting

```
DNS-Lookup Off
```

HostName - Specify the fully qualified domain name or IP address for the server

Use this directive to specify the domain name or an IP address returned to clients from document requests. If you specify a domain name, a domain name server must be able to resolve the name into an IP address. If you specify an IP address, the domain name server is not needed or accessed.

Example

```
HostName name or IP address
```

Initial Configuration File Setting: The name you specified for **Host name** during installation. The installation default is the default host name defined in your domain name server. If you remove this directive, the default host name defined in your domain name server is used.

imbeds - Specify whether server-side includes will be dynamically imbedded

Use this directive to specify if you want server-side include processing to be performed for documents served from the file system, CGI programs, or both. Server-side include processing is done on documents with a content type of `text/x-ssi-html`. Optionally, you can specify that server-side include processing also be done for documents with a content type of `text/html`. For more information about content types, see “AddType - Specify the data type of files with particular extensions” on page 95.

You can use server-side includes to dynamically insert information, such as the date, the size of a file, the last change date of a file, CGI or server-side include environment variables, or text documents into the document being returned. For more information on using server-side includes, see “Using server-side includes to insert information into CGI programs and HTML documents” on page 178.

Server-side include processing causes the server to search your documents for special commands each time they are served. This can affect the server's performance and slow down response time to clients.

The format of this directive is:

```
imbeds source [type]
```

source can be:

| | |
|--------------|--|
| on | Server-side include processing is done for documents from the file system and from CGI programs. |
| files | Server-side include processing is only done for documents from the file system. |
| cgi | Server-side include processing is only done for documents returned by CGI programs. |
| off | Server-side include processing is not done for any documents. |

The server checks the content type of each file it retrieves and the output of each CGI program it processes.

Server-side include processing is normally done only for documents having a content type of `text/x-ssi/html`. However, you can specify that documents with a content type of `text/html` be processed for server-side includes.

type can be:

| | |
|----------------|---|
| SSIOnly | Server-side include processing is done for documents with a content type of <code>text/x-ssi-html</code> . |
| html | Server-side include processing is done for documents with a content type of <code>text/html</code> and a content type of <code>text/x-ssi-html</code> . |

Basic

Important migration note: In previous releases, you could specify that documents with a specific extension, such as .html, .htm, or .shtml, be processed for server-side includes. In this release, the server treats html, .html, and .htm as **html**. Anything else is treated as **SSIOnly**.

Each extension must have an AddType directive defined with the correct content type. If you use extensions other than .htm or .html, make sure an AddType directive is defined with a content type of text/x-ssi/html.

Initial configuration file setting

```
imbeds on SSIOnly
```

Program default setting

```
imbeds off SSIOnly
```

Port - Specify the port on which you want the server to listen for requests

Use this directive to specify the port number the server should listen to for requests. The standard port number for HTTP is 80. Other port numbers less than 1024 are reserved for other TCP/IP applications and should not be used. Common ports used for proxy Web servers are 8080 and 8008.

When a port other than 80 is used, clients are required to include a specific port number on requests to the server. The port number is preceded by a colon and placed after the host name on the URL. For example, from the browser, the URL

```
http://www.turfco.com:8008/
```

requests the default welcome page from a host named www.turfco.com that is listening on port 8008.

You can use the -p option on the httpd command to override this setting when starting the server.

If you change this directive, you must stop your server and then start it again for the change to take effect. The server will not pick up the change if you only restart it.

Example

```
Port 8080
```

Initial configuration file setting: The number you specified for Port during installation. The installation default is 80.

ServerRoot - Specify the directory where the server program is installed

Use this directive to specify the directory where the server program is installed.

Example

```
ServerRoot    d:\webserve\BIN
```

Initial configuration file setting: The directory name you specified for **Executables directory** during installation. The installation default is:

```
ServerRoot    C:\WWW\BIN
```

Note: PASS and EXEC rules may be independent of this directory.

Directories and Welcome Page - Set viewing options

Use the directives described in this section to control how your server responds to requests containing a directory name. You can have the server search the directory for a welcome file to return, or you can have the server generate a directory listing.

By default, the server first looks for a welcome file. If no welcome file is present, the server displays a directory listing. Configuration settings control how directory listings appear and the icons that the listings use.

The server provides a set of default icons to use for directory listings. You can replace these icons with others using some of the directives described in this section.

AddBlankIcon - Specify the icon URL used to align the heading of directory listings

Use this directive to specify an icon to use for aligning the heading on directory listings. This can either be a blank icon or another icon you want to appear on the headings of your directory listings. For proper alignment, the icon you use must be the same size as the other icons you are using on your directory listings. The format of the directive is:

```
AddBlankIcon  icon-URL  alternate-text
```

Directories and Welcome Page

icon-URL

The last part of the URL for the icon. The server adds this value to the value of the `IconPath` directive to form the complete URL request. If the request is for a local file, the server translates the request through the mapping directives. For the icon to be retrieved, the mapping directives must allow the request to be passed.

If you are using the server as a proxy, the complete request must be a fully qualified URL pointing to your server.

alternate-text

The alternate text to use for the icon if the requesting browser is not displaying graphics.

Example

```
AddBlankIcon logo.gif logo
```

Initial configuration file setting

```
AddBlankIcon blank.gif
```

The default does not specify alternative text since the icon is blank.

AddDirIcon - Specify the icon URL for directories on directory listings

Use this directive to specify an icon for representing a directory on a directory listing. The format of the directive is:

```
AddDirIcon icon-URL alternate-text
```

icon-URL

The last part of the URL for the icon. The server adds this value to the value of the `IconPath` directive to form the complete URL request. If the request is for a local file, the server translates the request through the mapping directives. For the icon to be retrieved, the mapping directives must allow the request to be passed.

If you are using the server as a proxy, the complete request must be a fully qualified URL pointing to your server. You must map the URL to a local file and make sure that the mapping directives allow the URL to be passed.

alternate-text

The alternate text to use for the icon if the requesting browser is not displaying graphics.

Example

```
AddDirIcon direct.gif DIR
```

Initial configuration file setting

```
AddDirIcon dir.gif DIR
```

AddIcon - Bind an icon to a MIME content-type or encoding-type

Use this directive to specify icons for representing files with a specific MIME content-type or encoding-type. The server uses the icons on directory listings. The format of the directive is:

```
AddIcon icon-URL alternate-text type-template
```

icon-URL

The last part of the URL for the icon. The server adds this value to the value of the IconPath directive to form the complete URL request. If the request is for a local file, the server translates the request through the mapping directives. For the icon to be retrieved, the mapping directives must allow the request to be passed.

If you are using the server as a proxy, the complete request must be a fully qualified URL pointing to your server. You must map the URL to a local file and make sure that the mapping directives allow the URL to be passed.

alternate-text

The alternate text to use for the icon if the requesting browser is not displaying graphics.

type-template

Either a MIME content-type or encoding-type template. Content-type templates always contain a slash. Encoding-type templates never have a slash.

Example

```
AddIcon movie.gif video video/*
```

Initial configuration file setting

```
AddIcon binary.gif      BIN      binary
AddIcon text.gif         TXT      text/*
AddIcon image.gif        IMG      image/*
AddIcon movie.gif        MOV      video/*
AddIcon sound.gif        AU       audio/*
AddIcon tar.gif           AR       multipart/*tar
AddIcon compress.gif     MP       x-compress x-gzip
```

AddParentIcon - Specify the icon URL for a parent directory on directory listings

Use this directive to specify an icon for representing a parent directory on a directory listing. The format of the directive is:

```
AddParentIcon icon-URL alternate-text
```

Directories and Welcome Page

icon-URL

The last part of the URL for the icon. The server adds this value to the value of the IconPath directive to form the complete URL request. If the request is for a local file, the server translates the request through the mapping directives. For the icon to be retrieved, the mapping directives must allow the request to be passed.

If you are using the server as a proxy, the complete request must be a fully qualified URL pointing to your server. You must map the URL to a local file and make sure that the mapping directives allow the URL to be passed.

alternate-text

The alternate text to use for the icon if the requesting browser is not displaying graphics.

Example

```
AddParentIcon parent.gif UP
```

Initial configuration file setting

```
AddParentIcon back.gif UP
```

AddUnknownIcon - Specify the icon URL for unknown file types on directory listings

Use this directive to specify an icon for representing files with an unknown file type on a directory listing. The format of the directive is:

```
AddUnknownIcon icon-URL alternate-text
```

icon-URL

The last part of the URL for the icon. The server adds this value to the value of the IconPath directive to form the complete URL request. If the request is for a local file, the server translates the request through the mapping directives. For the icon to be retrieved, the mapping directives must allow the request to be passed.

If you are using the server as a proxy, the complete request must be a fully qualified URL pointing to your server. You must map the URL to a local file and make sure that the mapping directives allow the URL to be passed.

alternate-text

The alternate text to use for the icon if the requesting browser is not displaying graphics.

Example

```
AddUnknownIcon saywhat.gif huh
```

Initial configuration file setting

```
AddUnknownIcon unknown.gif ???
```

AlwaysWelcome - Specify if a welcome file is returned for all directory requests

Use this directive to specify if you want your server to always handle directory requests by first searching the directory for a welcome file.

The default value is `On`, which means that the server always searches the directory for a welcome file. The `Welcome` directive specifies the names of the files that the server recognizes as welcome files.

If you change the value to `Off`, the server first checks the last character of directory requests for the slash (`/`) character. If a directory request ends with a slash, the server searches the directory for a welcome file. If a directory request does not end with a slash, the server attempts to return a directory listing.

If the server does not find a welcome file, or `AlwaysWelcome` is set to `Off` and the directory request does not end in a slash, the `DirAccess` directive controls whether or not the server responds to the request with a directory listing.

Note: Setting `AlwaysWelcome` to `Off` does not affect requests that contain only your server name without a directory name. The server will always handle these requests by looking in your document root directory for a welcome file. The server cannot generate a directory listing for the document root directory.

Example

```
AlwaysWelcome Off
```

Initial configuration file setting

```
AlwaysWelcome On
```

DirAccess - Control directory listings

Use this directive to specify whether you want your server to return directory listings when requested. The values on the `Welcome` and `AlwaysWelcome` directives determine when a request is interpreted as a request for a directory listing.

The default value is `On`, which means that the server can return directory listings for all directories and subdirectories. If you want to control which directories and subdirectories the server can return directory listings for, use:

```
DirAccess Selective
```

If you change the value to `Off`, the server will not return directory listings.

If you change the value to `Selective`, the server will return directory listings for any directory that contains a file named `.www_browsable`. The contents of the `.www_browsable` file are not important; the server only checks for its existence.

Directories and Welcome Page

Examples:

```
DirAccess Off  
DirAccess Selective
```

Initial configuration file setting

```
DirAccess On
```

DirReadme - Control directory README files

Use this directive to specify if and where you want your server to display directory listing README files.

The default value is `Top`, which means that when the server returns a directory listing, it searches the directory for a file named `README`. If `README` is found, the server puts the contents of the file at the top of the directory listing.

If you change the value to `Bottom`, the server searches for a `README` file, but puts the contents at the bottom of the directory listing.

If you change the value to `Off`, the server does not search the directory for a `README` file.

Examples:

```
DirReadme Bottom  
DirReadme Off
```

Initial configuration file setting

```
DirReadme Top
```

DirShowBrackets - Use brackets around alternative text on directory listings

Use this directive to specify whether you want the server to put brackets around alternative text on directory listings. The directives that specify directory listing icons also contain alternate text. The alternate text is used in place of an icon if the requesting browser is not displaying graphics.

Example

```
DirShowBrackets Off
```

Initial configuration file setting

```
DirShowBrackets On
```

DirShowBytes - Show byte count for small files on directory listings

Use this directive to specify whether directory listings should include the exact byte count for files smaller than 1 KB.

A value of `Off` means the directory listing shows a size of 1 KB for all files that are 1 KB or smaller.

Example

```
DirShowBytes On
```

Initial configuration file setting

```
DirShowBytes Off
```

DirShowCase - Use case when sorting files on directory listings

Use this directive to specify whether directory listings should distinguish between uppercase and lowercase letters when sorting file names.

A value of `On` means uppercase letters are placed before lowercase letters.

Example

```
DirShowCase Off
```

Initial configuration file setting

```
DirShowCase On
```

DirShowDate - Show date last modified on directory listings

Use this directive to specify whether directory listings should include the last modification date for each file.

Example

```
DirShowDate Off
```

Initial configuration file setting

```
DirShowDate On
```

Directories and Welcome Page

DirShowDescription - Show descriptions for files on directory listings

Use this directive to specify whether directory listings should include descriptions for HTML files. The descriptions are taken from the files' HTML <title> tags.

Example

```
DirShowDescription Off
```

Initial configuration file setting

```
DirShowDescription On
```

DirShowHidden - Show hidden files on directory listings

Use this directive to specify whether directory listings should include any hidden files on the directory.

The server considers the following to be hidden files:

- Any file with the hidden attribute turned on
- Any file that has a name beginning with a period (.)

Example

```
DirShowHidden Off
```

Initial configuration file setting

```
DirShowHidden On
```

DirShowIcons - Show icons in directory listings

Use this directive to specify whether you want your server to include icons in directory listings. Icons can be used to provide a graphic representation of the content type of the files in the listing. The icons themselves are defined by the AddBlankIcon, AddDirIcon, AddIcon, AddParentIcon, and AddUnknownIcon directives.

Example

```
DirShowIcons Off
```

Initial configuration file setting

```
DirShowIcons On
```

DirShowMaxDescrLength - Set the maximum description length on directory listings

Use this directive to set the maximum number of characters to show in the description field on directory listings.

Example

```
DirShowMaxDescrLength 30
```

Initial configuration file setting

```
DirShowMaxDescrLength 25
```

DirShowMaxLength - Set the maximum length for file names on directory listings

Use this directive to set the maximum number of characters that will be used for file names on directory listings.

Example

```
DirShowMaxLength 30
```

Initial configuration file setting

```
DirShowMaxLength 25
```

DirShowMinLength - Set the minimum length for file names on directory listings

Use this directive to set the minimum number of characters that will always be reserved for file names on directory listings. File names in the directory can exceed this number. However, file names cannot be longer than the number specified on the DirShowMaxLength directive.

Example

```
DirShowMinLength 10
```

Initial configuration file setting

```
DirShowMinLength 15
```

Directories and Welcome Page

DirShowSize - Show file size on directory listings

Use this directive to specify whether directory listings should include the size of each file.

Example

```
DirShowSize Off
```

Initial configuration file setting

```
DirShowSize On
```

IconPath - Specify the path for the directory listing internal icons

Use this directive to specify URL information to be added at the beginning of each *icon-URL* specified on an `AddBlankIcon`, `AddDirIcon`, `AddParentIcon`, `AddUnknownIcon`, or `AddIcon` directive. The value you specify on this directive is added to the *icon-URL* value on each of the other directives to form the full request for the icon. The full request can be mapped to a file on your server or it can be a request to another server.

Attention: This directive must be before any of the other icon directives (`AddBlankIcon`, `AddDirIcon`, `AddParentIcon`, `AddUnknownIcon`, and `AddIcon`).

Example

```
IconPath http://icon.server.com:8080/httpd-internal-icons/
```

In the above example, each request for a directory list icon generates a request to a server named `icon.server.com`.

```
IconPath /icons/
```

With the default settings, each request for a directory list icon generates a request that begins with `/icons/`. The server uses its mapping rules to map the request to a local file.

Welcome - Specify names of welcome files

Use this directive to specify the name of a welcome file the server should look for to respond to requests that do not contain a specific file name. You can build a list of welcome files by putting multiple occurrences of this directive in the configuration file.

For requests that do not contain a file name or a directory name, the server always looks in the document root directory for a file that matches a name specified on a `Welcome` directive. If a match is found, the file is returned to the requester.

Directories and Welcome Page

For requests that contain a directory name but not a file name, the AlwaysWelcome directive controls whether the server looks in the directory for a welcome file to return. By default, AlwaysWelcome is set to a value of 0n. This means the server always looks in the requested directory for a file matching a name specified on a Welcome directive. If a match is found, the file is returned to the requester.

If the server finds more than one match between files in a directory and file names on Welcome directives, the order of the Welcome directives determines which file is returned. The server uses the Welcome directive closest to the top of the configuration file.

If the server does not find a welcome file in the directory, the DirAccess directive controls whether or not the server responds to the request with a directory listing.

The format of the Welcome directive is:

```
Welcome file-name [Server-IP-address or hostname]
```

file-name

A file name you want to define as being a welcome file.

Server-IP-address or *hostname*

If you are using multiple IP addresses or virtual hosts, use this parameter to specify an IP address or a host name. (For more information on using multiple IP addresses or virtual hosts, see “Running your server with multiple IP addresses or virtual hosts” on page 19.) The server uses the directive only for requests that come to the server on this IP address or for this host. For an IP address, this is the address of the server’s network connection, not the address of the requesting client.

You can specify an IP address (for example, 204.146.167.72) or you can specify a host name (for example, hostA.bcd.com).

This parameter is optional. Without this parameter, the server uses the directive for all requests regardless of the IP address the requests come in on or the host name in the URLs.

Important migration note: Beginning with Version 4.2, a wildcard character can no longer be specified for a server’s IP address.

Examples:

```
Welcome letsgo.html  
Welcome Welcome.html
```

The above example defines two welcome pages and assumes the AlwaysWelcome directive is set to its default of 0n. For requests that do not contain a file name, the server would try to return a welcome file from the directory specified on the request (or document root directory if the request does not specify a file name or a directory). The server would first look for a file named letsgo.html. If the directory does not have a letsgo.html file, the server would look for a file named Welcome.html.

Directories and Welcome Page

```
Welcome CustomerA.html 9.67.106.79
Welcome CustomerB.html 9.83.100.45
```

Your server would look for different welcome files based on the IP address of the network connection the request comes in on. For requests coming in on 9.67.106.79 the server would look for welcome files named CustomerA.html. For requests coming in on 9.83.100.45, the server would look for welcome files named CustomerB.html. If the request comes in on a different IP address, the server looks for the default address.

```
Welcome CustomerA.html hostA.bcd.com
Welcome CustomerB.html hostB.bcd.com
```

Your server would look for different welcome files based on the host name in the URL. For requests coming in for hostA, the server would look for welcome files named CustomerA.html. For requests coming in for hostB, the server would look for welcome files named CustomerB.html. If the request comes in for a different host, the server looks for the default host name.

Initial configuration file setting

```
Welcome Welcome.html
Welcome welcome.html
Welcome index.html
Welcome Frntpage.html
```

The above default values are shown in the order used by the default configuration.

User directories

Use the directives described in this section to control whether individual users of your server can have their own private Web documents.

HomeDir - Specify the directory that contains user subdirectories

Use this directive to specify the name of a directory on which you want to allow individual users to have their own subdirectories. Each user can have a subdirectory off of the directory name you specify. URL requests to user subdirectories are in the format `~username`. The server passes these requests to the appropriate user subdirectory on the HomeDir directory.

You can use the HomeDir directive either by itself or together with the UserDir directive. Whether or not you use UserDir determines how the server maps URL requests to the user subdirectories.

Directories and Welcome Page

If you use HomeDir without UserDir, all files within the user subdirectories are accessible to clients. See the description of the UserDir directive for a description and example of using HomeDir with UserDir.

Note: By using this directive, you are telling the server to process all requests to user subdirectories. You do not need to put any additional Pass directives in the configuration file.

Example

```
HomeDir c:\user
```

In the above example, the server would map URL requests containing Γ username to the c:\user\username directory.

Initial Configuration File Setting: None. By default the server does not accept requests to user subdirectories.

UserDir - Specify the name of the accessible subdirectory off of user subdirectories

Use this directive to specify the name of a subdirectory where individual users can keep the files they want to be accessible to clients of the server. This directive limits client access to just one subdirectory within each user subdirectory. Each individual subdirectory off of the HomeDir directory can have a subdirectory with the name you specify.

To use this directive, you must also use the HomeDir directive.

For example, you might want to use this directive if your server has access to a LAN server where users have home directories. You can use this directive to define one subdirectory off of each user directory where users could keep Web pages they want to be accessible.

Example

```
HomeDir c:\user  
UserDir html_doc
```

In the above example the server would map URL requests containing Γ username to the c:\user\username\html_doc directory.

Initial Configuration File Setting: None

Logging and Reporting

Logging and Reporting - Customize access and error logs and generate access reports

Use the directives described in this section to control your server's logs. You can have the server log incoming requests and server errors. If your server is running as a caching proxy server, you can log access requests for files from the proxy server's cache.

You can specify the path and file where you want these logs to be kept, how requests should be logged, and which requests you do not want to log.

You can view the access, error, and trace logs from the server window. Although by default the access log will not write to the server window by default to ensure better server performance, you can change this by setting the LogToGUI directive to On.

You most likely will want to use the common log format. This is the default format and it is the same format used by most other types of Web servers. If you plan to use the access log report function, you are required to use the common log format.

AccessLog - Name the path for the access log file

Use this directive to specify the path and file name where you want the server to log access statistics. By default, the server writes an entry to this log each time a client sends the server a request. You can use the NoLog directive if you do not want to log requests from certain clients. For a description of the NoLog directive, refer to "NoLog - Suppress log entries for specific hosts or domains matching a template" on page 71.

The server starts a new log file each day at midnight if it is running. Otherwise, the server starts a new log file the first time you start it on a given day. When creating the file, the server uses the file name you specify and appends a date extension. The date extension is in the format *Mmmddyyyy*, where *Mmm* is the first three letters of the month; *dd* is the day of the month; and *yyyy* is the year.

It is a good idea to remove old log files, because they can take up a significant amount of space on your hard drive. For information about removing old log files, refer to "AccessLogArchive - Remove existing access, agent, or referer log files or run a user exit" on page 55.

Example

```
AccessLog c:\server\logs\accesslog
```

Logging and Reporting

Initial configuration file setting

```
AccessLog c:\path\httpd-log
```

`c:\path` is the value you entered for **Logs directory** at installation. The installation default is `c:\WWW\LOGS`

AccessLogArchive - Remove existing access, agent, or referer log files or run a user exit

Values specified on the `AccessLogArchive` directive apply to access, agent, and referer logs. The collective size includes the size of all access logs or all agent logs or all referer logs, not the collective size of all types of logs.

At midnight each night, the server closes the current log and creates a new log file for the coming day. You can choose to do one of the following actions with the closed logs:

- Remove log files of a given age or when a given amount of storage is used by the collection of log files
- Allow closed logs to remain on your hard drive
- Branch to a user exit

To remove access, agent, or referer logs of a given age, specify this directive, in addition to the `AccessLogExpire` directive. To remove logs when their collective size exceeds a certain amount of storage, specify this directive, in addition to the `AccessLogSizeLimit` directive.

To allow closed logs to remain on your hard drive, you can accept the default, which is `AccessLogArchive none`.

To branch to a user exit, specify the path to the user exit and any parameters for the user exit on the `AccessLogArchive` directive. The server will append to this directive the path to the access, agent, or referer log.

The `AccessLogArchive` directive can be specified in any of the following formats:

```
AccessLogArchive  purge
AccessLogArchive  none
AccessLogArchive  userexit path_to_the_user-exit_program [parameters for the user-exit]
```

purge

Remove access log files of a given age or when their collective size exceeds a given amount of storage.

none

Do not remove access log files. `none` is the default.

Logging and Reporting

userexit

Specifies the path of the user-exit program you want to branch to. You can optionally specify the parameters for your user-exit program, as shown in the following examples. The server appends the path to the access log to the directive.

Examples:

```
AccessLogArchive purge
AccessLogArchive none
AccessLogArchive userexit c:\accback\backup.exe -d -a
```

For the `AccessLogArchive userexit` example, the user exit invocation is:

```
AccessLogArchive userexit c:\accback\backup.exe -d -a c:\www\logs\httpd-log
```

Initial configuration file setting

```
AccessLogArchive none
```

Program default setting: None.

AccessLogExcludeURL - Suppress log entries for specific files or directories

Use this directive to specify that you do not want to log access requests made for specific files or directories that match a given URL template. For example, you might not want to log access requests for GIF files or you might not want to log access requests to a particular file or directory on your server.

You can have multiple occurrences of this directive in your configuration file. You can also put multiple entries for the same directive if you separate them by one or more spaces.

Example

```
AccessLogExcludeURL *.gif
AccessLogExcludeURL /Freebies/*
```

Initial configuration file setting: None. The server includes in the access log requests for all files and directories.

Program default setting: None.

AccessLogExcludeMethod - Suppress log entries for files or directories requested by a given method

Use this directive to specify that you do not want to log access requests made for files or directories by using a specific method. For example, you might not want to log DELETE requests for files or directories.

You can have multiple occurrences of this directive in your configuration file. You can also put multiple methods on the same directive if you separate them by one or more spaces.

Example

```
AccessLogExcludeMethod GET  
AccessLogExcludeMethod PUT  
AccessLogExcludeMethod POST  
AccessLogExcludeMethod DELETE
```

Initial configuration file setting: None. The server includes in the access log the files and directories requested by all types of methods.

Program default setting: None.

AccessLogExcludeMimeType - Suppress log entries for specific MIME types

Use this directive to specify that you do not want to log access requests made for directories or files of a given MIME type. (Examples of MIME types are text/html, image/gif, and image/jpeg.) For example, you might not want to log access requests for GIF images.

You can have multiple occurrences of this directive in your configuration file. You can also put multiple MIME types on the same directive if you separate them by one or more spaces.

Example

```
AccessLogExcludeMimeType image/gif
```

Initial configuration file setting: None. The access log includes requests to the server for files and directories of all MIME types.

Program default setting: None.

Logging and Reporting

AccessLogExcludeReturnCode - Suppress log entries for specific return codes

Use this directive to specify that you do not want to log access requests that fall within a given range of error code numbers. These error code numbers are httpd status codes. You cannot specify individual codes. Specifying 300 indicates that you want to exclude access requests with redirection return codes (301, 302, 303, and 304). To exclude these requests, you would specify:

```
AccessLogExcludeReturnCode 300
```

You can have multiple occurrences of this directive in your configuration file. You can also put multiple return codes on the same directive if you separate them by one or more spaces.

Example

```
AccessLogExcludeReturnCode 300
```

Initial configuration file setting: None. The access log includes all requests to the server, regardless of the code.

Program default setting: None.

AccessLogExpire - Remove existing access log files when they reach a given age in days

Use this directive to specify that you want to remove access log files when they reach a certain age (in days).

This directive requires that you also specify the `AccessLogArchive` directive, described under “`AccessLogArchive - Remove existing access, agent, or referer log files or run a user exit`” on page 55. You can have only one occurrence of this directive in your configuration file.

The format of the `AccessLogExpire` directive is:

```
AccessLogExpire number-of-days
```

number-of-days

Specifies that access logs older than this value are to be removed.

number-of-days must be an integer; decimal values such as 1.5 are not valid. The default is 0, a value that indicates that no expiration date exists.

The file creation date, as reported by the operating system, is used to determine the age of the access log file. The suffix of the filename, such as `httpd-log.Mar221996.extension`, is not used to determine file age. (*extension* is the file extension.)

Example

```
AccessLogExpire 10
```

Initial configuration file setting

```
AccessLogExpire 0
```

Program default setting: None.

AccessLogSizeLimit - Remove existing access log files when they reach a given collective size

Use this directive to specify that you want to remove access log files when they reach a collective size (in megabytes).

This directive requires that you also specify the `AccessLogArchive` directive, described under “`AccessLogArchive - Remove existing access, agent, or referer log files or run a user exit`” on page 55. You can have only one occurrence of this directive in your configuration file.

The format of the `AccessLogSizeLimit` directive is:

```
AccessLogSizeLimit number-of-megabytes
```

number-of-megabytes

Specifies that when the combined size of the access log files exceeds this value, files are deleted starting with the oldest file, until the collective size is within the limit specified on the `AccessLogSizeLimit` directive. *number-of-megabytes* must be an integer. The default is 0, a value that indicates that no access log files are to be removed.

This directive takes effect after the `AccessLogExpire` directive has taken effect.

Example

```
AccessLogSizeLimit 4
```

Initial configuration file setting

```
AccessLogSizeLimit 0
```

Program default setting: None.

AccessReportDescription - Give a short description of the report to be created

Use this directive to include a short description of the report to be created with this template.

Logging and Reporting

Example

```
AccessReportDescription Report on Web page accesses
```

Initial configuration file setting: None.

Program default setting: None.

AccessReportExcludeURL - Suppress from the report the log entries for specific files or directories

Use this directive to specify that you do not want to include in the access report requests made for specific files or directories that match a given URL template. For example, you might not want to include in the report access requests for GIF files or you might not want to include access requests to a particular file or directory on your server.

You can have multiple occurrences of this directive in your configuration file.

Example

```
AccessReportExcludeURL *.gif  
AccessReportExcludeURL oldfiles*
```

Initial configuration file setting: None.

Program default setting: None.

AccessReportIncludeURL - Include in the report only log entries for specific files or directories

Use this directive to specify that you want to include in the access report only access requests made for specific files or directories that match a given URL template. For example, you might want to include in the report only access requests for HTML files or you might want to include access requests to a particular file or directory on your server.

You can have multiple occurrences of this directive in your configuration file.

Example

```
AccessReportIncludeURL /*.html
```

Initial configuration file setting: None.

Program default setting: None.

AccessReportExcludeHostName - Suppress from the report the log entries for specific host names

Use this directive to specify that you do not want to include in the access report requests made by host names or IP addresses that match a given template.

You can have multiple occurrences of this directive in your configuration file.

Note: To exclude host names, you must set the DNS-Lookup directive to On. If the DNS-Lookup directive is set to Off (the default), you can exclude IP addresses only.

Example

```
AccessReportExcludeHostName 9.85.*.*  
AccessReportExcludeHostName *.edu
```

Initial configuration file setting: None.

Program default setting: None.

AccessReportIncludeHostName - Include in the report only log entries for specific host names

Use this directive to specify that you want to include in the access report requests made by host names or IP addresses that match a given template.

You can have multiple occurrences of this directive in your configuration file.

Note: To include host names, you must set the DNS-Lookup directive to On. If the DNS-Lookup directive is set to Off (the default), you can include IP addresses only.

Example

```
AccessReportIncludeHostName 9.9.99.*  
AccessReportIncludeHostName *.com
```

Initial configuration file setting: None.

Program default setting: None.

Logging and Reporting

AccessReportExcludeMethod - Suppress from the report the log entries of a given method type

Use this directive to specify that you do not want to include in the access report requests of a given method type.

You can have multiple occurrences of this directive in your configuration file.

Example

```
AccessReportExcludeMethod GET
AccessReportExcludeMethod PUT
AccessReportExcludeMethod POST
```

Initial configuration file setting: None.

Program default setting: None.

AccessReportExcludeReturnCode - Suppress from the report the log entries with a given return code

Use this directive to specify that you do not want to include in the access report requests that fall within a given range of error code numbers. These error code numbers are httpd status codes. You cannot specify individual return codes. Specifying 300 indicates that you want to exclude from the report access requests with redirection return codes (301, 302, 303, and 304). To exclude these requests from the report, you would specify `AccessReportExcludeReturnCode 300`

You can have multiple occurrences of this directive in your configuration file.

Syntax:

```
AccessReportExcludeReturnCode return code
```

Example

```
AccessReportExcludeReturnCode 200
AccessReportExcludeReturnCode 400
```

Initial configuration file setting: None.

Program default setting: None.

AccessReportRoot - Name the path for the root directory where access log reports are stored

Use this directive to specify the path and file name where you want the server to store access log reports and summary databases. We recommend that you accept the default path. If you choose to specify a different path, you will need to create the new directory with all the appropriate permissions and add a PASS directive to enable the server to honor requests for reports in that directory.

If you are running with workload management enabled, you should have unique AccessReportRoot directives for each instance of httpd based on the subsystem name. If you specify -SN system1, you should have AccessReportRoot /usr/lpp/internet/server_root/pub/reports/system1 and a corresponding Pass directive.

Example

```
AccessReportRoot C:\WWW\BIN\REPORTS
```

Initial configuration file setting None.

Program default setting: None.

AccessReportTemplate - Name the report template

Use this directive to specify the name of the report template. There is a default template named "Top50." For more information, see "Tailoring the reports your server creates" on page 161.

The format of the AccessReportTemplate is:

```
AccessReportTemplate report_title
```

report_title

The name of the report. The name cannot include any blanks.

Example

```
AccessReportTemplate PageHits
```

Initial configuration file setting

```
AccessReportTemplate Top50 {
AccessReportDescription Top 50 most frequently requested files
                        and most frequent visitors
AccessReportTopList 50
}
```

Program default setting: None.

Logging and Reporting

AccessReportTopList - Specify the top number of items on

which to report

Use this directive to specify the top number of items on which to report.

The format of the AccessReportTopList is:

```
AccessReportTopList top_number|all
```

top_number

Specifies that the report is to include the *top_number* most frequently occurring entries in the access log. This must be an integer value.

all Specifies that the report is to include all entries in the report.

Example

```
AccessReportTopList 10
```

Initial configuration file setting: None.

Program default setting: None.

AgentLog - Name the path for the agent log file

Use this directive to specify the path and file name where you want the server to log statistics about which Web browser was used to access a Web page. By default the server writes an entry to this log each time a client sends the server a request. For every entry made in the access log, the agent log has a corresponding entry that indicates the browser used to display the page or file requested by the client.

The server starts a new agent log file each day at midnight if it is running. Otherwise, the server starts a new log file the first time you start it on a given day. When creating the file, the server uses the file name you specify and appends a date extension. The date extension is in the format *Mmmddyyyy*, where *Mmm* is the first three letters of the month; *dd* is the day of the month; and *yyyy* is the year.

Example

```
AgentLog c:\server\logs\agent-log
```

Initial configuration file setting

```
AgentLog c:\path\agent-log
```

c:\path is the value you entered for **Logs directory** at installation. The installation default is c:\WWWLOGS

Program default setting: None.

CacheAccessLog - Specify the path for the cache access log files

If the server is running as a proxy, you can log requests to the cache separately from other requests. Use the `CacheAccessLog` directive to specify the path and file name where you want the server to put access requests for cached files. To enable logging of requests to the proxy cache, the following directives must be defined:

- Caching must be turned ON (default is OFF)
- `CacheRoot` (by default, no `CacheRoot` is defined)
- `CacheAccessLog`

The value of `CacheAccessLog` can either be an absolute path or a path relative to `ServerRoot` (one example is shown of each).

Note: If you choose to use `CacheAccessLog`, access requests for cached files are logged, but they are not included in the access reports. Access reports contain only information from access logs, not from cache access logs. Therefore, if you want access reports to contain access requests for cached files, do *not* specify the `CacheAccessLog` directive.

The server starts a new log file each day at midnight if it is running. Otherwise, the server starts a new log file the first time you start it on a given day. When creating the file, the server uses the file name you specify and appends a date extension. The date extension is in the format `Mmmddyyyy`, where `Mmm` is the first three letters of the month; `dd` is the day of the month; and `yyyy` is the year.

Note: It is a good idea to occasionally check the path that contains your log files. If your server has a large amount of activity, your log files could grow to the point that they cause your server to run out of disk space. Make sure the files are not taking up too much space and delete the log files you no longer need.

The format of this directive is

```
CacheAccessLog <file_path>
```

Example

```
CacheAccessLog c:\absolute\path\logfile
CacheAccessLog c:\logs\logfile
```

Initial configuration file setting: None. The server does not log cache access requests unless you include this directive in your configuration file.

Program default setting: None.

Logging and Reporting

CgiErrorLog - Name the path for the CGI error log file

Use this directive to specify the path and file name where you want the server to log standard error output (stderr) from CGI programs.

The server starts a new CGI error file each day at midnight if it is running. Otherwise, the server starts a new log file the first time you start it on a given day. When creating the file, the server uses the file name you specify and appends a date extension. The date extension is in the format *Mmmddyyyy*, where *Mmm* is the first three letters of the month; *dd* is the day of the month; and *yyyy* is the year.

Example

```
CgiErrorLog c:\server\logs\cgi-error
```

Initial configuration file setting

```
CgiErrorLog c:\path\cgi-error
```

c:\path is the value you entered for **Logs directory** at installation. The installation default is *c:\WWWLOGS*

Program default setting: None.

ErrorLog - Name the file where you want to log internal server errors

Use this directive to specify the path and file name where you want the server to log internal errors.

The server starts a new log file each day at midnight if it is running. Otherwise, the server starts a new log file the first time you start it on a given day. When creating the file, the server uses the file name you specify and appends a date extension. The date extension is in the format *Mmmddyyyy*, where *Mmm* is the first three letters of the month; *dd* is the day of the month; and *yyyy* is the year.

Note: If your server has a large amount of activity, your log files could grow to the point that they cause your server to run out of disk space. To prevent the files from taking up too much space, you can specify the `ErrorLogArchive` directive described under “`ErrorLogArchive - Remove existing error or CGI error log files or run a user exit`” on page 67.

Example

```
ErrorLog c:\server\logs\errorlog
```

Logging and Reporting

Initial configuration file setting

```
ErrorLog c:\path\httpd-error
```

`c:\path\` is the value you entered for **Logs directory** at installation. The installation default is `c:\WWW\LOGS`

Program default setting: None.

ErrorLogArchive - Remove existing error or CGI error log files or run a user exit

Values specified on the `ErrorLogArchive` directive apply to error and CGI error logs. The collective size includes the size of all error logs or all CGI logs, not the collective size of both types of logs.

At midnight each night, the server closes the current error and CGI error logs and creates a new log files for the coming day. You can choose to do one of the following actions with the closed error logs:

- Remove log files of a given age or when a given amount of storage is used by the collection of error log files
- Allow closed logs to remain on your hard drive
- Branch to a user exit.

To remove logs of a given age, specify this directive, in addition to the `ErrorLogExpire` directive. To remove logs when their collective size exceeds a certain amount of storage, specify this directive, in addition to the `ErrorLogSizeLimit` directive.

To allow closed logs to remain on your hard drive, you can accept the default, which is `ErrorLogArchive none`.

To branch to a user exit, specify the path to the user exit and any parameters for the user exit on the `ErrorLogArchive` directive. The server will append to this directive the path to the error or CGI error log.

The `ErrorLogArchive` directive can be specified in any of the following formats:

```
ErrorLogArchive  purge
ErrorLogArchive  none
ErrorLogArchive  userexit path_to_the_user-exit program [parameters for the
user-exit]
```

purge

Remove log files of a given age or when their collective size exceeds a given amount of storage.

Logging and Reporting

none

Do not remove log files. `none` is the default.

userexit

Specifies the path of the user-exit program you want to branch to. You can optionally specify the parameters for your user-exit program, as shown in the following examples. The server appends the path to the error log to the directive.

Examples:

```
ErrorLogArchive purge
ErrorLogArchive none
ErrorLogArchive userexit c:\errback\backup.exe -d -a
```

For the `ErrorLogArchive userexit` example, the user exit invocation is:

```
ErrorLogArchive userexit c:\errback\backup.exe -d -a c:\www\logs\httpd-error
```

Initial configuration file setting

```
ErrorLogArchive none
```

Program default setting: None.

ErrorLogExpire - Remove existing error log files when they reach a given age in days

Use this directive to specify that you want to remove error log files when they reach a certain age (in days).

This directive requires that you also specify the `ErrorLogArchive` directive, described under “`ErrorLogArchive - Remove existing error or CGI error log files or run a user exit`” on page 67. You can have only one occurrence of this directive in your configuration file.

The format of the `ErrorLogExpire` directive is:

```
ErrorLogExpire number-of-days
```

number-of-days

Specifies that error logs older than this value are to be removed. *number-of-days* must be an integer; decimal values such as 1.5 are not valid. The default is 0, a value that indicates that no expiration date exists.

The file creation date, as reported by the operating system, is used to determine the age of the error log file. The suffix of the filename, such as `httpd-log.Mar221996`, is not used to determine file age.

Example

```
ErrorLogExpire 10
```

Logging and Reporting

Initial configuration file setting

```
ErrorLogExpire 0
```

Program default setting: None.

ErrorLogSizeLimit - Remove existing error log files when they reach a given collective size

Use this directive to specify that you want to remove error log files when they reach a collective size (in megabytes).

This directive requires that you also specify the `ErrorLogArchive` directive, described under “`ErrorLogArchive - Remove existing error or CGI error log files or run a user exit`” on page 67. You can have only one occurrence of this directive in your configuration file.

The format of the `ErrorLogSizeLimit` directive is:

```
ErrorLogSizeLimit number-of-megabytes
```

number-of-megabytes

Specifies that when the sum total size of the error log files exceeds this value, files are deleted starting with the oldest file, until the collective size is within the limit specified on the `ErrorLogSizeLimit` directive. *number-of-megabytes* must be an integer. The default is 0, a value that indicates that no error log files are to be removed.

This directive takes effect after the `ErrorLogExpire` directive has taken effect.

Example

```
ErrorLogSizeLimit 4
```

Initial configuration file setting

```
ErrorLogSizeLimit 0
```

Program default setting: None.

LogFormat - Specify common or old log file format

Use this directive to specify whether you want your server to write log files in the common format or old format.

If you plan to use the reporting functions described under “Tailoring the reports your server creates” on page 161, you must accept the default file format, common.

Logging and Reporting

The common format is the one used by most Web servers. The old format was used with early versions of Web servers from CERN. You most likely will want to use the common format, which is the default.

Example

```
LogFormat Old
```

Initial configuration file setting

```
LogFormat Common
```

Program default setting: Common.

LogTime - Specify GMT or local time stamps in log files

Use this directive to specify whether your logs should record entries using local time or Greenwich Mean Time (GMT).

Example

```
LogTime GMT
```

Initial configuration file setting

```
LogTime LocalTime
```

Program default setting: LocalTime.

LogToGUI - Specify whether access log writes to GUI

Use this directive to specify whether you want your server to write access log files to the server window (also known as graphical user interface or GUI). The default setting of OFF improves server performance.

Example

```
LogToGUI On
```

Initial configuration file setting

```
LogToGUI Off
```

Program default setting: Off.

NoLog - Suppress log entries for specific hosts or domains matching a template

Use this directive to specify that you do not want to log access requests made from specific hosts or domains that match a given template. For example, you may not want to log access requests from local hosts.

You can have multiple occurrences of this directive in your configuration file. You can also put multiple templates on the same directive if you separate them by one or more spaces. You can use host names or IP number addresses on the templates.

Note: To use host name templates, you must set the DNS-Lookup directive to 0n. If the DNS-Lookup directive is set to 0ff (the default), you can use IP address templates only.

Example

```
NoLog 128.141.* *.edu localhost.*
```

Initial configuration file setting: None

Program default setting: None.

RefererLog - Name the path for the referer log file

Use this directive to specify the path and file name where you want the server log the identity of the Web page that referred to (linked to) the requested Web page. By default the server writes an entry to this log each time a client sends the server a request. For every entry made in the access log, the referer log has a corresponding entry that indicates which page referred to the page that was requested by the client. If no page referred to the requested page, the entry is two quotation marks (" ").

The server starts a new referer log file each day at midnight if it is running. Otherwise, the server starts a new log file the first time you start it on a given day. When creating the file, the server uses the file name you specify and appends a date extension. The date extension is in the format *Mmmddyyyy*, where *Mmm* is the first three letters of the month; *dd* is the day of the month; and *yyyy* is the year.

Example

```
RefererLog c:\server\logs\referer-log
```

Initial configuration file setting

```
RefererLog c:\path\referer-log
```

Logging and Reporting

c:\path is the value you entered for **Logs directory** at installation. The installation default is c:\WWWLOGS

Program default setting: None.

ReportDataCompressionProgram - Specify path to the compression program

Use this directive to specify the path to the compression program (such as PKZIP2, GZIP, or compress) and any program parameters for the compression program. Include any command line parameters on the same line. This compression program is to be used to compress access data log files.

Example

```
ReportDataCompressionProgram c:\bin\pkunzip2
```

Initial configuration file setting

```
ReportDataCompressionProgram
```

Program default setting: None.

ReportDataUnCompressionProgram - Specify path to the uncompression program

Use this directive to specify the path to the uncompression program (such as UNZIP, GZIP, or uncompress) and any program parameters for the uncompression program. Include any command line parameters on the same line. This uncompression program is to be used to uncompress access data log files.

Example

```
ReportDataUnCompressionProgram c:\bin\pkunzip2
```

Initial configuration file setting

```
ReportDataUnCompressionProgram
```

Program default setting: None.

ReportDataCompressionSuffix - Specify the suffix appended to the compressed file

Use this directive to specify the suffix appended to the compressed file.

Example

```
ReportDataCompressionSuffix .zip
```

Initial configuration file setting

```
ReportDataCompressionSuffix
```

Program default setting: None.

ReportProcessOldLogs - Check for old logs in the log directory

Use this directive to indicate that you want the server to check for old access logs in the log directory that are not listed in the list of log files that have been processed into reports. With this directive, you can process old access log files by:

- Appending the data from the old access log files to existing reports
- Creating reports for all access log files and overwriting existing reports
- Creating a report for the most recently created access log file.

The format of the ReportProcessOldLogs directive is:

```
ReportProcessOldLogs append|force|last
```

append

Add to existing access log reports data from log files that were not originally included in the reports.

force

Overwrite existing access log reports with reports based on data from all access log files, regardless of whether they were originally included in the reports.

Note: The only way to erase reports named *access.mmdyyyy* files is to archive them with the ReportDataArchive directive, described under “ReportDataArchive - Specify whether to remove existing accessdata files” on page 74.

last

Create reports based on data from the most recently created access log file.

Examples

```
ReportProcessOldLogs append
ReportProcessOldLogs force
ReportProcessOldLogs last
```

Logging and Reporting

Initial configuration file setting

ReportProcessOldLogs append

Program default setting: None.

ReportDataSizeLimit - Remove existing access data files when they reach a given collective size

Use this directive to specify that you want to remove access data files when they reach a collective size (in megabytes).

This directive requires that you also specify the ReportDataArchive directive, described under “AccessLogArchive - Remove existing access, agent, or referer log files or run a user exit” on page 55. You can have only one occurrence of this directive in your configuration file.

The format of the ReportDataSizeLimit directive is:

```
ReportDataSizeLimit number-of-megabytes
```

number-of-megabytes

Specifies that when the sum total size of the access data files exceeds this value, files are deleted starting with the oldest file, until the collective size is within the limit specified on the ReportDataSizeLimit directive. *number-of-megabytes* must be an integer. The default is 0, a value that indicates that no access data files are to be removed.

This directive takes effect after the ReportDataExpire directive has taken effect.

Example

```
ReportDataSizeLimit 4
```

Initial configuration file setting

```
ReportDataSizeLimit 0
```

Program default setting: None.

ReportDataArchive - Specify whether to remove existing accessdata files

Use this directive to specify whether you want to remove existing access data log files.

If you want to remove access data files, you also need to specify the ReportDataExpire directive, described under “ReportDataExpire - Remove existing access data files when

Logging and Reporting

they reach a given age in days” on page 76. You can have only one occurrence of this directive in your configuration file.

Even after you remove access data files, the data from these files is still available for reports to use, until you specify the ReportProcessOldLogs directive with the force option.

The ReportDataArchive directive can be specified in any of the following formats:

```
ReportDataArchive purge
ReportDataArchive none
ReportDataArchive userexit path_to_the_user-exit_program [parameters for the
user-exit]
```

purge

Remove access data files of a given age or when their collective size exceeds a given amount of storage.

none

Do not remove access data files. none is the default.

userexit

Specifies the path of the user-exit program you want to branch to. You can optionally specify the parameters for your user-exit program, as shown in the following examples. The server appends the path to the access log to the directive.

Examples:

```
ReportDataArchive purge
ReportDataArchive none
ReportDataArchive userexit c:\accback\backup.exe -d -a
```

For the ReportDataArchive userexit example, the user exit invocation is:

```
ReportDataArchive userexit c:\accback\backup.exe -d -a c:\www\logs\httpd-log
```

Example

```
ReportDataArchive purge
```

Initial configuration file setting

```
ReportDataArchive none
```

None

Program default setting: None.

Access control

ReportDataExpire - Remove existing access data files when they reach a given age in days

Use this directive to specify that you want to remove access log data files when they reach a certain age (in days).

This directive requires that you also specify the ReportDataArchive directive, described under “ReportDataArchive - Specify whether to remove existing accessdata files” on page 74. You can have only one occurrence of this directive in your configuration file.

```
ReportDataExpire number-of-days
```

number-of-days

Specifies that reports older than this value are to be removed. *number-of-days* must be an integer; decimal values such as 1.5 are not valid. The default is 0, a value that indicates that no expiration date exists.

The file creation date, as reported by the operating system, is used to determine the age of the error log file. The suffix of the filename, such as httpd-log.Mar221996, is not used to determine file age.

Example

```
ReportDataExpire 10
```

Initial configuration file setting

```
ReportDataExpire 0
```

Program default setting: None.

Access control - Set up access control for the server

Use the directives described in this section to control access to your server’s resources.

You link protection setups to groups of files based on the requests that are used to access those files. Use the DefProt and Protect directives to define the requests you want to protect.

You can define the actual protection setup in a separate protection statement or directly in the configuration file. Within the configuration file, you can define and label a protection setup using the Protection directive. You can also define a protection setup directly on a DefProt or Protect directive.

This section also describes the subdirectives that define a protection setup.

See Chapter 7, “Protecting your server” on page 201 for step-by-step instructions on protecting your server resources.

DefProt - Specify default protection setup for requests that match a template

Use this directive to associate a default protection setup with requests that match a template.

Attention: For protection to work properly, you must put your DefProt and Protect directives before any Pass or Exec directives in your configuration file.

The format of the directive is:

```
DefProt request-template setup [FOR Server-IP-address or hostname]
```

request-template

A template for requests that you want to associate with a default protection setup. The server compares incoming client requests to the template and associates a protection setup if there is a match.

Protection is not actually activated for requests matching the template unless the request also matches a template on a subsequent Protect directive. See the description of the Protect directive for an explanation of how it is used with DefProt.

setup

The default protection setup you want to associate with requests that match *request-template*. Protection setup is defined with protection subdirectives. See “Protection Subdirectives” on page 84 for descriptions of the protection subdirectives. This parameter can take one of three forms:

- A full path and file name identifying a separate file that contains the protection subdirectives.
- A protection setup label name that matches a name defined earlier on a Protection directive. The Protection directive contains the protection subdirectives.
- The actual protection subdirectives. The subdirectives must be enclosed in braces {}. The left brace character must be the last character on the same line as the DefProt directive. Each subdirective follows on its own line. The right brace character must be on its own line following the last subdirective line. You cannot put any comment lines between the braces.

FOR *Server-IP-address* or *hostname*

If you are using multiple IP addresses or virtual hosts, use this parameter to specify an IP address or a host name. (For more information on using multiple IP addresses or virtual hosts, see “Running your server with multiple IP addresses or

Access control

virtual hosts” on page 19.) The server uses the directive only for requests that come to the server on this IP address or for this host. For an IP address, this is the address of the server’s network connection, not the address of the requesting client.

You can specify an IP address (for example, FOR 204.146.167.72) or you can specify a host name (for example, FOR hostA.bcd.com).

This parameter is optional. Without this parameter, the server uses the directive for all requests regardless of the IP address the requests come in on or the host name in the URL.

Notes:

1. To use this parameter, the *setup* parameter must be in the form of a path and file name or a protection setup label. You cannot use protection subdirectives enclosed in braces for the *setup* parameter.
2. To use this parameter, you must put **FOR**, or some other character string (without blanks), between the *setup* parameter and the *IP-address* or *hostname*.

Important migration note: Beginning with Version 4.2, a wildcard character can no longer be specified for a server’s IP address.

Examples:

```
DefProt /secret/* d:\server\protect\setup1.acc
DefProt /secret/* SECRET-PROT
```

The above example uses a label name to point to the protection subdirectives. The label name must match a label name on a Protection directive. The Protection directive must come before the DefProt directive.

```
DefProt {
  AuthType Basic
  ServerID restricted
  PasswdFile d:\docs\WWW\restrict.pwd
  GroupFile d:\docs\WWW\restrict.grp
  GetMask authors
  PutMask authors
}
```

The above example includes the protection subdirectives as part of the DefProt directive.

```
DefProt /secret/* CustomerA-PROT 9.67.106.79
DefProt /secret/* CustomerB-PROT 9.83.100.45
```

The above examples use the optional IP address parameter. If your server receives requests that begin with */secret/*, it associates a different default protection setup with the request based on the IP address of the network connection the request comes in on. For requests coming in on 9.67.106.79, the server associates the request with default protection defined on a Protection directive with a label of CustomerA-PROT.

Access control

For requests coming in on 9.83.100.45, the server associates the request with default protection defined on a Protection directive with a label of CustomerB-PROT.

```
DefProt /secret/* CustomerA-PROT hostA.bcd.com
DefProt /secret/* CustomerB-PROT hostB.bcd.com
```

The above examples use the optional host name parameter. If your server receives requests that begin with /secret/, it associates a different default protection setup with the request based on the host name in the URL. For requests coming in for hostA, the server associates the request with default protection defined on a Protection directive with a label of CustomerA-PROT. For requests coming in on hostB, the server associates the request with default protection defined on a Protection directive with a label of CustomerB-PROT.

Program default setting: None.

Initial configuration file setting: None.

Protect - Activate protection setup for requests that match a template

Use this directive to activate protection setup rules for requests that match a template.

Attention: For protection to work properly, you must put your DefProt and Protect directives before any Pass or Exec directives in your configuration file.

The format of the directive is different depending upon whether you want to point to a label or file containing the protection subdirectives or you want to include the protection subdirectives as part of the Protect directive.

If you want to point to a label or file containing the protection subdirectives, the format is as follows:

```
Protect request-template [setup-file/label [FOR Server-IP-address or hostname ]]
```

If you want to include the protection subdirectives as part of the Protect directive, the format is as follows:

```
Protect request-template [Server-IP-address or hostname] {
    subdirective value
    subdirective value
    .
    .
    .
}
```

request-template

A template for requests that you want to activate protection for. The server compares incoming client requests to the template and activates protection if there is a match.

Access control

setup-file/label

If you are pointing to a label or file containing the protection subdirectives, use this parameter to identify the protection setup you want to activate for requests that match *request-template*.

This parameter is optional. If it is omitted, the protection setup is defined by the most recent DefProt directive that contains a matching template.

Protection setup is defined with protection subdirectives. See “Protection Subdirectives” on page 84 for descriptions of the protection subdirectives. If present, this parameter can take one of three forms:

- A full path and file name identifying a separate file that contains the protection subdirectives.
- A protection setup label name that matches a name defined earlier on a Protection directive. The Protection directive contains the protection subdirectives.

subdirective value

If you want to include the protection subdirectives as part of the Protect directive, use this parameter. The left brace character must be the last character on the same line as the Protect directive. Each subdirective follows on its own line. The right brace character must be on its own line following the last subdirective line. You cannot put any comment lines between the braces.

See “Protection Subdirectives” on page 84 for descriptions of the protection subdirectives.

FOR *Server-IP-address* or *hostname*

If you are using multiple IP addresses or virtual hosts, use this parameter to specify an IP address or a host name. (For more information on using multiple IP addresses or virtual hosts, see “Running your server with multiple IP addresses or virtual hosts” on page 19.) The server uses the directive only for requests that come to the server on this IP address or for this host. For an IP address, this is the address of the server’s network connection, not the address of the requesting client.

You can specify an IP address (for example, FOR 204.146.167.72) or you can specify a host name (for example, FOR hostA.bcd.com).

This parameter is optional. Without this parameter, the server uses the directive for all requests regardless of the IP address the requests come in on or the host name in the URL.

Notes:

1. To use this parameter, you must also use the *setup-file/label* parameter or the *subdirective value* parameters.
2. To use this parameter with a *setup-file/label* parameter, you must put **FOR**, or some other character string (without blanks), between the *setup-file/label* parameter and the *IP-address* or *hostname* parameter.

Access control

3. To use this parameter with *subdirective value* parameters, do NOT include **FOR** before the *IP-address* or *hostname*.

Important migration note: Beginning with Version 4.2, a wildcard character can no longer be specified for a server's IP address.

Examples:

```
Protection BUS-PROT {
  AuthType Basic
  ServerID restricted
  PasswdFile d:\docs\WWW\restrict.pwd
  GroupFile d:\docs\WWW\restrict.grp
  GetMask authors
  PutMask authors
}
DefProt /secret/*      d:\server\protect\setup1.acc
Protect /secret/scoop/*
Protect /secret/business/*  BUS-PROT
Protect /topsecret/* {
  AuthType Basic
  ServerID restricted
  PasswdFile d:\docs\WWW\restrict.pwd
  GroupFile d:\docs\WWW\restrict.grp
  GetMask topbrass
  PutMask topbrass
}
Pass  /secret/scoop/*      d:\WWW\restricted\*
Pass  /secret/business/*   d:\WWW\confidential\*
Pass  /topsecret/*         d:\WWW\topsecret\*
```

In the above example, the server activates protection as follows:

- Requests that start with `/secret/scoop/` activate protection. The protection setup is defined in the `d:\server\protect\setup1.acc` protection setup file. Since the `Protect` directive does not specify a protection setup, the protection setup on the previously matching `DefProt` directive is used.
- Requests beginning with `/secret/business/` activate protection. The protection setup is defined on the `Protection` directive that has a label of `BUS-PROT`.
- Requests beginning with `/topsecret/` activate protection. The protection setup is included directly on the `Protect` directive.

Access control

```
Protect /secret/* CustomerA-PROT FOR 9.67.106.79
Protect /secret/* CustomerB-PROT FOR 9.83.100.45
Protect /topsecret/* 9.67.106.79 {
    AuthType Basic
    ServerID restricted
    PasswdFile /docs/WWW/customer-A.pwd
    GroupFile /docs/WWW/customer-A.grp
    GetMask A-brass
    PutMask A-brass
}
Protect /topsecret/* 9.83.100.45 {
    AuthType Basic
    ServerID restricted
    PasswdFile /docs/WWW/customer-B.pwd
    GroupFile /docs/WWW/customer-B.grp
    GetMask B-brass
    PutMask B-brass
}
```

The above examples use IP addresses. If your server receives requests that begin with `/secret/` or `/topsecret/`, it activates a different protection setup for the request based on the IP address of the network connection the request comes in on.

For `/secret/` requests coming in on 9.67.106.79, the server activates the protection setup defined on a Protection directive with a label of `CustomerA-PROT`. For `/topsecret/` requests coming in on 9.67.106.79, the server activates the protection setup defined inline on the Protect directive for `/topsecret/`.

For `/secret/` requests coming in on 9.83.100.45, the server activates the protection setup defined on a Protection directive with a label of `CustomerB-PROT`. For `/topsecret/` requests coming in on 9.83.100.45, the server activates the protection setup defined inline on the Protect directive for `/topsecret/`.

```
Protect /secret/* CustomerA-PROT FOR hostA.bcd.com
Protect /secret/* CustomerB-PROT FOR hostB.bcd.com
Protect /topsecret/* hostA.bcd.com {
    AuthType Basic
    ServerID restricted
    PasswdFile /docs/WWW/customer-A.pwd
    GroupFile /docs/WWW/customer-A.grp
    GetMask A-brass
    PutMask A-brass
}
Protect /topsecret/* hostB.bcd.com {
    AuthType Basic
    ServerID restricted
    PasswdFile /docs/WWW/customer-B.pwd
    GroupFile /docs/WWW/customer-B.grp
    GetMask B-brass
    PutMask B-brass
}
```

Access control

The above examples use virtual hosts. If your server receives requests that begin with `/secret/` or `/topsecret/`, it activates a different protection setup for the request based on the host name in the URL.

For `/secret/` requests coming in for `hostA.bcd.com`, the server activates the protection setup defined on a Protection directive with a label of `CustomerA-PROT`. For `/topsecret/` requests coming in for `hostA.bcd.com`, the server activates the protection setup defined inline on the Protect directive for `/topsecret/`.

For `/secret/` requests coming in for `hostB.bcd.com`, the server activates the protection setup defined on a Protection directive with a label of `CustomerB-PROT`. For `/topsecret/` requests coming in for `hostB.bcd.com`, the server activates the protection setup defined inline on the Protect directive for `/topsecret/`.

Default: Protection is provided for the Configuration and Administration forms by a Protect directive with a request template of `/admin-bin/*`.

Protection - Define a named protection setup within the configuration file

Use this directive to define a protection setup within the configuration file. You give the protection setup a name and define the type of protection using protection subdirectives.

Note: In the configuration file, you must place Protection directives before any DefProt or Protect directives that point to them.

The format of the directive is:

```
Protection label-name {  
    subdirective value  
    subdirective value  
    .  
    .  
    .  
}
```

label-name

The name you want to associate with this protection setup. The name can then be used by subsequent DefProt and Protect directives to point to this protection setup.

subdirective value

Put a protection subdirective and its value on each line between the left brace and the right brace. You cannot put any comment lines between the braces.

See “Protection Subdirectives” on page 84 for descriptions of the protection subdirectives.

Example

Access control

```
Protection NAME-ME {
    AuthType Basic
    ServerID restricted
    PasswdFile d:\WWW\password.pwd
    GroupFile d:\WWW\group.grp
    GetMask groupname
    PutMask groupname
}
```

Initial configuration file setting

```
Protection PROT-ADMIN {
    PasswdFile C:\TCPIP\ETC\ADMIN.PWD
    Mask All@(*)
    PostMask All@(*)
    PutMask All@(*)
    GetMask All@(*)
    AuthType Basic
    ServerID Private_Authorization
}
```

Protection Subdirectives

Following are descriptions of each of the protection subdirectives that can be used in a protection setup. The subdirectives are in alphabetical order.

Protection setups can either be in separate files or within the configuration file as part of DefProt, Protect, or Protection directives.

See “Protection example (without SSL client authentication)” on page 218 and the previous descriptions of the DefProt, Protect, and Protection directives for examples of using protection setups.

ACLOverride - Specify that ACL files override protection setups

Use this subdirective with a value of 0n if you want Access Control List files (ACL) to override the masks specified in the protection setup. If a directory being protected by the protection setup has an ACL file, the mask subdirectives in the protection setup are ignored. (The mask subdirectives are DeleteMask, GetMask, Mask, PostMask, and PutMask.)

See “Using Access Control List (ACL) files” on page 215 for more information on ACL files.

Example

```
ACLOverride 0n
```

AuthType - Specify authentication type

Use this subdirective when limiting access based on user names and passwords. Specify the type of authentication to use when the client sends a password to the server. With basic authentication (`AuthType Basic`), passwords are sent to the server as plain text. They are encoded, but not encrypted.

Example

```
AuthType Basic
```

DeleteMask - Specify the user names, groups, and addresses allowed to delete files

Use this subdirective to specify user names, groups, and addresses templates authorized to make DELETE requests to a protected directory. See “Rules for specifying user names, group names, and address templates” on page 210.

Example

```
DeleteMask authors,(niceguy,goodie)@96.96.3.1,128.141.*.*
```

GetMask - Specify the user names, groups, and addresses allowed to get files

Use this subdirective to specify user names, groups, and address templates authorized to make GET requests to a protected directory. See “Rules for specifying user names, group names, and address templates” on page 210.

Example

```
GetMask authors,(niceguy,goodie)@96.96.3.1,128.141.*.*
```

GroupFile - Specify the location of the associated group file

Use this subdirective to specify the path and file name of the server group file that you want this protection setup to use. The groups defined within the server group file can then be used by:

- Any mask subdirectives that are part of the protection setup. (The mask subdirectives are `DeleteMask`, `GetMask`, `Mask`, `PostMask`, and `PutMask`.)
- Any ACL file on a directory that is protected by the protection setup.

See “Using server group files” on page 214 more information about server group files.

Access control

Example

```
GroupFile d:\docs\WWW\restrict.group
```

Mask - Specify the user names, groups, and addresses allowed to make HTTP requests

Use this subdirective to specify user names, groups, and address templates authorized to make HTTP requests not covered by other mask subdirectives. See “Rules for specifying user names, group names, and address templates” on page 210. See “Methods - Set method acceptance” on page 118 for descriptions of the HTTP methods supported by the server.

Example

```
Mask authors,(niceguy,goodie)@96.96.3.1,128.141.*.*
```

PasswdFile - Specify the location of the associated password file

Use this subdirective when limiting access based on user names and passwords. Specify the path and name of the password file that you want this protection setup to use.

Because some browsers such as NetScape cache userid/password by security realm (ServerId) within host, follow these guidelines when specifying ServerId and password files:

- Protection setups that use the same password file should use the same ServerId.
- Protection setups that use different password files should use different ServerIds.

Examples:

```
PasswdFile c:\WWW\restrict.password
```

PostMask - Specify the user names, groups, and addresses allowed to post files

For a secure server, use this subdirective to specify users, groups, and address templates authorized to make POST requests to a protected directory. See “Rules for specifying user names, group names, and address templates” on page 210.

Example

```
PostMask authors,(niceguy,goodie)@96.96.3.1,128.141.*.*
```

PutMask - Specify the users names, groups, and addresses allowed to put files

Use this subdirective to specify users, groups, and address templates authorized to make PUT requests to a protected directory. See “Rules for specifying user names, group names, and address templates” on page 210.

Example

```
PutMask authors,(niceguy,goodie)@96.96.3.1,128.141.*.*
```

ServerID - Specify a name to associate with the password file

Use this subdirective when limiting access based on user names and passwords. Specify a name you want to associate with the password file being used. The name does not need to be a real machine name.

The name is used as an identifier to the requester. Since different protection setups can use different password files, having a name associated with the protection setup can help the client decide which password to send. Most clients display this name when prompting for a user name and password.

Because some browsers such as NetScape cache userid/password by security realm (ServerId) within host, follow these guidelines when specifying ServerId and password files:

- Protection setups that use the same password file should use the same ServerId.
- Protection setups that use different password files should use different ServerIds.

Example

```
ServerID restricted
```

SSL client authentication subdirectives

If you implement SSL client authentication, the server requests the client's certificate when the client makes an **https** request. The server establishes a secure connection whether or not the client has a valid certificate.

You can restrict who can access documents by using password files and/or user or group authentication in protection setups as described in Chapter 7, “Protecting your server” on page 201. You can further restrict who can access documents by coding SSL client authentication parameters on protection setups, ACL files, or both. Coding SSL client authentication parameters on protection setups is described in “Creating

Access control

protection setups for SSL client authentication” on page 212; coding them in ACL files is described in “Step 5. Limiting access to individual files” on page 214.

Using SSL client authentication parameters as subdirectives, you can specify that the client certificate is valid or you can specify all or part of the Distinguished Name (DN) of a client or of the certification authority (CA) who issued the client's certificate.

When you use SSL client authentication parameters, the server first checks to see if the client certificate is valid, as is. If not, it compares any DN information in a protection setup and then compares any DN information in an ACL file with the DN information in the client's certificate. If the DN information matches, the server serves the document.

The following can be specified on the Protection or Protect directive:

- The validity of the client certificate.
 - `SSL_ClientAuth client` - indicates that the client certificate is valid without verifying any of the Distinguished Name information in the client certificate. Only the keyword ***client*** is valid with this parameter.
- All or any of the following parameters that make up a client's Distinguished Name in the client's certificate:
 - `CommonName` - the client's common name
 - `Country` - the country in which the client resides
 - `Locality` - the locality in which the client resides
 - `StateOrProvince` - the state or province in which the client resides
 - `Organization` - the organization of the client
 - `OrgUnit` - the organizational unit of the client
- All or any of the following subdirectives that make up the CA's Distinguished Name in the client's certificate:
 - `IssuerCommonName` - the CA's common name
 - `IssuerCountry` - the country in which the CA resides
 - `IssuerLocality` - the locality in which the CA resides
 - `IssuerStateOrProvince` - the state or province in which the CA resides
 - `IssuerOrganization` - the organization of the CA
 - `IssuerOrgUnit` - the organizational unit of the CA

Example

```
Protect /topsecret/* {  
  CommonName "Dr Sheila A. Jones"  
  Organization "RTP Quick Care Center"  
  Mask Anybody@(*)  
}
```

In the above example of an inline Protect directive, any **https** request beginning with `/topsecret/` causes the server to request the client's certificate. If the client's common name is Dr Sheila A. Jones and the client's organization is RTP Quick Care Center, then the server returns the document to the client.

Hints and tips for coding SSL client authentication parameters

- Specify any or all of a client or CA's DN
- Enclose DN information that contains blanks in double quotes (as shown in the above example).
- Make sure the DN information matches the DN information in the client's certificate. This information is case sensitive and must have the same punctuation.
- Do not use wildcard characters for any of the parameters.

Security - Set up network security for the server

For a secure server, use the directives described in this section to control how your server uses network security functions.

If you change these directives, you must stop your server and then start it again for the changes to take effect. The server will not pick up the changes if you only restart it.

KeyFile - Set name for key ring file

Use this directive to set the name for the key ring file.

If you change this directive, you must stop your server and then start it again for the change to take effect. The server will not pick up the change if you only restart it.

Initial configuration file setting

KeyFile keyfile.kyr

Program default setting

None

NormalMode - Turn port on or off for HTTP

Use this directive to turn on or off the port defined by the Port directive.

Set NormalMode on for an HTTP connection. If you also want an SSL connection, set SSLMode on.

Security

Note: You should have either this directive or the SSLMode directive turned on. However, if both NormalMode and SSLMode are turned off, the server will start in normal mode, and you won't have a secure network connection.

If you change this directive, you must stop your server and then start it again for the change to take effect. The server will not pick up the change if you only restart it.

Initial configuration file setting

```
NormalMode on
```

Program default setting

```
NormalMode on
```

SSLClientAuth - Enable SSL client authentication

Use this directive to enable SSL client authentication.

When this directive is enabled, the server requests a certificate from each client that makes an **https** request. Only enable client authentication if you need to authenticate clients; SSL client authentication increases network traffic.

If you change this directive, you must stop your server and then start it again for the change to take effect. The server will not pick up the change if you only restart it.

Initial configuration file setting

```
SSLClientAuth off
```

Program default setting

```
SSLClientAuth on
```

SSLMode - Turn port on or off for SSL

Use this directive to turn on or off the port defined by the SSLPort directive.

Set SSLMode on for an SSL connection. If you also want an HTTP connection, set NormalMode on.

Note: You should have either this directive or the NormalMode directive turned on. However, if both NormalMode and SSLMode are turned off, the server will start in normal mode, and you won't have a secure network connection.

If you change this directive, you must stop your server and then start it again for the change to take effect. The server will not pick up the change if you only restart it.

Initial configuration file setting

Multi-format processing

```
SSLMode on
```

Program default setting

```
SSLMode off
```

SSLPort - Set port for SSL security

Use this directive to set the port for SSL security. The server will use this port only for HTTPS requests. (Requests for HTTP will still come on the port that you set with the Port directive.)

If you want to use a port other than 443, specify a port above 1024.

If you change this directive, you must stop your server and then start it again for the change to take effect. The server will not pick up the change if you only restart it.

Initial configuration file setting

```
SSLPort 443
```

Program default setting

```
SSLPort 443
```

Multi-format processing - Define file extensions for multi-format processing

Use the directives described in this section to associate files with particular extensions to the meta-information found in the headers of incoming requests. Based on the file extension (suffixes) specified in these directives, the server binds files to a content type, content encoding, content language, character set, or to a browser sending a request.

Multi-Format Processing

Multi-format processing is only enabled when the requesting URL contains the .multi suffix or does not have a suffix (and a file with that name and no suffix does not exist).

The resource mapping directives, AddType, AddEncoding, AddLanguage, AddCharSet, and AddClient, are used to associate meta-information from request headers with file suffixes or extensions. Meta-information can consist of MIME type, encoding, quality, charset, language, and browser (agent) type.

Multi-format processing

The following table identifies meta-information that is associated with each directive.

| Directive | Suffix | MIME | Encoding | Quality | Charset | Language Agent |
|-------------|--------|------|----------|---------------|--------------|----------------|
| AddType | X | X | X | X (Optional) | X (Optional) | |
| AddEncoding | X | | X | X (Optional) | | |
| AddLanguage | X | | | X (Optional) | | X |
| AddCharSet | X | | | X (Optional) | X | |
| AddClient | X | | | 2.0 (Implied) | | X |

There are two pieces to multi-format processing:

- Request headers from the browser

The browser sends accept-headers containing acceptable values (content-type, content-encoding, language, charset) that you can associate to file suffixes with the configuration directives. The browser also sends a user-agent header that identifies its browser type that you can associate with file suffixes in the same manner.

- The requested URL

The server finds all files with any extension that matches the directory and file name and uses multi-format processing to choose the best file to return.

Computing file quality

Multi-format processing computes the quality of a file based on the set of suffixes in the directory for the requested file name and sends the highest quality it finds. A perfect match is always the highest quality.

Quality is a floating point number between 0.0 and 1.0 that represents the relative desirability of a file. For example, if the file abc.html is requested and cannot be found, the server searches the directory for all files that match abc.*. Multi-format processing adds the * and finds all that may qualify.

When computing file quality, each file is given the value of 1.0 and this is multiplied by the quality of each suffix. The quality of a suffix is based on the quality value specified in the directive which is multiplied by the quality value specified in the accept-header. This can be shown as:

$$\text{fileQ} = (1.0 \times ((\text{suffixQ1}) \times (\text{suffixQ2}) \dots))$$

or

$$\text{fileQ} = (1.0 \times ((\text{directiveQ} \times \text{headerQ}) \times (\text{directiveQ} \times \text{headerQ}) \dots))$$

Multi-format processing

The quality value is optional on many of the directives. If one is not specified, then the value of 1.0 is assumed. Also, if the accept-header does not specify a quality value, 1.0 is assumed.

If the server is processing a file and suffix that is defined in the list of AddClient directives, and the requester agent matches the defined string, the server doubles the file quality (implying a value of 2.0). If no match is found, this does not happen.

Example: If you have HTML source files in different languages,

```
myfile.du.html
myfile.uk.html
```

you could use these directives for multi-format processing:

```
AddLanguage .du du
AddLanguage .uk en_UK
AddType .html text/html 8bit
```

Then, if a browser sends a request for `myfile.multi` and sends the header `Accept-Language: du`, the server returns `myfile.du.html`. The server provides defaults for most commonly used extensions. Use the extension definition directives only if you need to add new definitions or change the default definitions.

AddLanguage - Specify the language of files with particular extensions

Use this directive to bind files with a particular extension to a language. The format of the directive is:

```
AddLanguage .extension language
```

.extension

The file extension pattern.

language

The language you want to bind to files that match the corresponding extension pattern.

Examples:

```
AddLanguage .en en_US
```

This example defines files with a `.en` extension as being in American English.

```
AddLanguage .uk en_UK
```

This example defines files with a `.uk` extension as being in United Kingdom English.

Initial configuration file setting: None.

Multi-format processing

AddEncoding - Specify the MIME content encoding of files with particular extensions

Use this directive to bind files with a particular extension to a MIME-encoding type. This directive is seldom used.

The format of the directive is:

```
AddEncoding .extension encoding
```

.extension

The file extension pattern.

encoding

The MIME-encoding type you want to bind to files that match the corresponding extension pattern.

Example

```
AddEncoding .qp quoted_printable
```

Initial configuration file setting

```
AddEncoding .Z x-compress 1.0  
AddEncoding .gz x-gzip 1.0
```

AddCharSet - Specify the character set documents are encoded in

Use this directive to specify the character set (code page) documents are stored in. (You can also specify character set on the AddType directive.)

The format of the directive is:

```
AddCharSet .extension character-set
```

.extension

The file extension pattern.

character-set

The character set you want to associate with text documents. For the documents that you assign a character set to, the server tells client browsers what character set to use when displaying the document. If you want to use the character-set field, you must also include a value for the quality field.

Example

```
AddCharSet .932 IBM-932
```

Initial configuration file setting: None

AddType - Specify the data type of files with particular extensions

Use this directive to bind files with a particular extension to a MIME type/subtype. You can have multiple occurrences of this directive in your configuration file. The server provides defaults for most commonly used extensions. The format of the directive is:

```
AddType .extension type/subtype encoding [quality[ character-set]]
```

.extension

The file extension pattern. You can use the wildcard character (*) only on the following two special extension patterns:

- *.* Matches all file names that contain a dot character (.) and have not been matched by other rules
- * Matches all file names that do not contain a dot character (.) and have not been matched by other rules

type/subtype

The MIME type and subtype you want to bind to files that match the corresponding extension pattern.

encoding

The MIME content encoding to which the data has been converted. In most cases, the appropriate encoding is 7bit, 8bit, or binary, and is determined as follows:

- 7bit** Data is all represented as short (less than 1000 characters) lines of 8859-1 ASCII data. Source code or plain text files usually fall into this category. Exceptions would be files containing line-drawing characters or accented characters.
- 8bit** Data is represented as short lines, but may contain characters with the high bit set (for example, line-drawing characters or accented characters). PostScript files and text files from European sites usually fall into this category.
- binary** This encoding can be used for all data types. Data may contain not only non-ASCII characters, but also long (greater than 1000 characters) lines. Almost every file of type image/*, audio/*, and video/* falls into this category, as do binary data files of type application/*.

Any other value of encoding receives the same treatment as binary and is passed in MIME headers as a content encoding MIME header. 7bit and 8bit is not sent in MIME headers.

quality

An optional indicator of relative value (on a scale of 0.0 to 1.0) for the content type. The quality value is used if multiple representations of a file are matched by a request. The server selects the file that is associated with the highest quality

Multi-format processing

value. For example, if the file `internet.ps` is requested, and the server has the following `AddType` directives:

```
AddType .ps application/postscript 8bit 1.0
AddType *.* application/binary      binary 0.3
```

the server would use the `application/postscript` line because its quality number is higher.

character-set

An optional indicator of the character set you want to associate with text documents. For the documents that you assign a character set to, the server tells client browsers what character set to use when displaying the document. If you want to use the `character-set` field, you must also include a value for the quality field.

Example

```
AddType .bin application/octet-stream binary 0.8
```

Defaults:

```
AddType .class application/octet-stream binary 1.0
AddType .mime www/mime                      binary 1.0
AddType .bin application/octet-stream binary 1.0
AddType .oda application/oda                binary 1.0
AddType .pdf application/pdf                binary 1.0
AddType .ai application/postscript          8bit 0.5
AddType .PS application/postscript          8bit 0.8
AddType .eps application/postscript          8bit 0.8
AddType .ps application/postscript          8bit 0.8
AddType .rtf application/x-rtf              7bit 1.0
AddType .csh application/x-csh              7bit 0.5
AddType .dvi application/x-dvi              binary 1.0
AddType .hdf application/x-hdf              binary 1.0
AddType .latex application/x-latex          8bit 1.0
AddType .nc application/x-netcdf            binary 1.0
AddType .cdf application/x-netcdf            binary 1.0
AddType .sh application/x-sh                7bit 0.5
AddType .tcl application/x-tcl              7bit 0.5
AddType .tex application/x-tex              8bit 1.0
AddType .texi application/x-texinfo         7bit 1.0
AddType .texinfo application/x-texinfo      7bit 1.0
AddType .t application/x-troff              7bit 0.5
AddType .roff application/x-troff           7bit 0.5
AddType .tr application/x-troff             7bit 0.5
AddType .man application/x-troff-man         7bit 0.5
AddType .me application/x-troff-me          7bit 0.5
AddType .ms application/x-troff-ms          7bit 0.5
AddType .src application/x-wais-source       7bit 1.0
AddType .bcpio application/x-bcpio          binary 1.0
AddType .cpio application/x-cpio            binary 1.0
AddType .gtar application/x-gtar            binary 1.0
AddType .shar application/x-shar            8bit 1.0
```

Multi-format processing

```
AddType .sv4cpio application/x-sv4cpio binary 1.0
AddType .sv4crc application/x-sv4crc binary 1.0
```

The following are neutral CAE formats:

```
AddType .igs application/iges binary 1.0
AddType .iges application/iges binary 1.0
AddType .IGS application/iges binary 1.0
AddType .IGES application/iges binary 1.0
AddType .stp application/STEP 8bit 1.0
AddType .STP application/STEP 8bit 1.0
AddType .step application/STEP 8bit 1.0
AddType .STEP application/STEP 8bit 1.0
AddType .dxf application/dxf binary 1.0
AddType .DXF application/dxf binary 1.0
AddType .vda application/vda binary 1.0
AddType .VDA application/vda binary 1.0
AddType .set application/set 8bit 1.0
AddType .SET application/set 8bit 1.0
AddType .stl application/SLA 8bit 1.0
AddType .STL application/SLA 8bit 1.0
```

The following are vendor-specific CAD-formats commonly used by CERN and in HEP institutes:

```
AddType .dwg application/acad binary 1.0
AddType .DWG application/acad binary 1.0
AddType .SOL application/solids binary 1.0
AddType .DRW application/drafting binary 1.0
AddType .prt application/pro_eng binary 1.0
AddType .PRT application/pro_eng binary 1.0
AddType .unv application/i-deas binary 1.0
AddType .UNV application/i-deas binary 1.0
AddType .CCAD application/clariscad binary 1.0
AddType .snd audio/basic binary 1.0
AddType .au audio/basic binary 1.0
AddType .aiff audio/x-aiff binary 1.0
AddType .aifc audio/x-aiff binary 1.0
AddType .aif audio/x-aiff binary 1.0
AddType .wav audio/x-wav binary 1.0
AddType .gif image/gif binary 1.0
AddType .ief image/ief binary 1.0
AddType .jpg image/jpeg binary 1.0
AddType .JPG image/jpeg binary 1.0
AddType .JPE image/jpeg binary 1.0
AddType .jpe image/jpeg binary 1.0
AddType .JPEG image/jpeg binary 1.0
AddType .jpeg image/jpeg binary 1.0
AddType .tif image/tiff binary 1.0
AddType .tiff image/tiff binary 1.0
AddType .ras image/cmu-raster binary 1.0
AddType .pnm image/x-portable-anymap binary 1.0
AddType .pbm image/x-portable-bitmap binary 1.0
```

Multi-format processing

| | | | | |
|---------|--------|---------------------------|--------|-----|
| AddType | .pgm | image/x-portable-graymap | binary | 1.0 |
| AddType | .ppm | image/x-portable-pixmap | binary | 1.0 |
| AddType | .rgb | image/x-rgb | binary | 1.0 |
| AddType | .xbm | image/x-xbitmap | 7bit | 1.0 |
| AddType | .xpm | image/x-xpixmap | binary | 1.0 |
| AddType | .xwd | image/x-xwindowdump | binary | 1.0 |
| AddType | .html | text/html | 8bit | 1.0 |
| AddType | .htm | text/html | 8bit | 1.0 |
| AddType | .htmls | text/html | 8bit | 1.0 |
| AddType | .c | text/plain | 7bit | 0.5 |
| AddType | .h | text/plain | 7bit | 0.5 |
| AddType | .C | text/plain | 7bit | 0.5 |
| AddType | .cc | text/plain | 7bit | 0.5 |
| AddType | .hh | text/plain | 7bit | 0.5 |
| AddType | .java | text/plain | 7bit | 0.5 |
| AddType | .m | text/plain | 7bit | 0.5 |
| AddType | .f90 | text/plain | 7bit | 0.5 |
| AddType | .txt | text/plain | 7bit | 0.5 |
| AddType | .rtx | text/richtext | 7bit | 1.0 |
| AddType | .tsv | text/tab-separated-values | 7bit | 1.0 |
| AddType | .etx | text/x-setext | 7bit | 0.9 |
| AddType | .MPG | video/mpeg | binary | 1.0 |
| AddType | .mpg | video/mpeg | binary | 1.0 |
| AddType | .MPE | video/mpeg | binary | 1.0 |
| AddType | .mpe | video/mpeg | binary | 1.0 |
| AddType | .MPEG | video/mpeg | binary | 1.0 |
| AddType | .mpeg | video/mpeg | binary | 1.0 |
| AddType | .qt | video/quicktime | binary | 1.0 |
| AddType | .mov | video/quicktime | binary | 1.0 |
| AddType | .avi | video/x-msvideo | binary | 1.0 |
| AddType | .movie | video/x-sgi-movie | binary | 1.0 |
| AddType | .gz | multipart/x-gzip | binary | 1.0 |
| AddType | .zip | multipart/x-zip | binary | 1.0 |
| AddType | .tar | multipart/x-tar | binary | 1.0 |
| AddType | .ustar | multipart/x-ustar | binary | 1.0 |
| AddType | *.* | www/unknown | binary | 0.2 |
| AddType | * | www/unknown | binary | 0.2 |
| AddType | .cxx | text/plain | 7bit | 0.5 |
| AddType | .for | text/plain | 7bit | 0.5 |
| AddType | .mar | text/plain | 7bit | 0.5 |
| AddType | .log | text/plain | 7bit | 0.5 |
| AddType | .com | text/plain | 7bit | 0.5 |
| AddType | .sdl | text/plain | 7bit | 0.5 |
| AddType | .list | text/plain | 7bit | 0.5 |
| AddType | .lst | text/plain | 7bit | 0.5 |
| AddType | .def | text/plain | 7bit | 0.5 |
| AddType | .conf | text/plain | 7bit | 0.5 |
| AddType | . | text/plain | 7bit | 0.5 |

SuffixCaseSense - Specify whether extension definitions are case sensitive

Use this directive to specify whether you want your server to distinguish between uppercase and lowercase letters when comparing file extensions to the extension patterns on AddClient, AddCharSet, AddType, AddEncoding, and AddLanguage directives. By default, the server does not distinguish between uppercase and lowercase.

Example

```
SuffixCaseSense On
```

Initial configuration file setting

```
SuffixCaseSense Off
```

AddClient - Specify file extensions for requesting clients

Use this directive to bind files with particular extensions to the type and version of client that is sending the request. This is often referred to as ***automatic browser detection***.

All HTTP requests contain a User-Agent header that identifies the browser sending the request. The Internet Connection Secure Server enables you to detect which browser was used to send a request and, based on this information, respond with a version of a Web page, a document, or other file that is appropriate for that browser.

For example, your server can send a page written in HTML 3.0 only to browsers that are known to support it and send a version of the same page written in HTML 2.0 to all other browsers.

Automatic browser detection is only effective for multi-format processing; that is when a requesting URL specifies a file without an extension and no file with that name exists, or it specifies a file with the extension **.multi**. For example, a link from this HTML anchor tag initiates multi-format processing:

```
<A HREF="http://www.raleigh.ibm.com/cjld/tscores.multi">
```

As a result, the server will evaluate the values passed in the request headers (such as User-Agent), along with the extensions of all the tscores files and the associations specified in the directives. Based on this, it will try to find the file that is the 'best match' to send in its response.

You can have multiple occurrences of this directive in your configuration file. The sequence of AddClient directives is important. The first AddClient directive that

Multi-format processing

matches a client's User-Agent value is the one that will be used to determine the file extension.

If a client's User-Agent is not matched in an AddClient directive, the server looks for a generic file extension (.htm or .html) to send. If the server cannot find a generic file extension, it uses an algorithm to calculate the quality of all the extensions for that file and sends the file whose extension yields the highest quality, considering it to be the 'best match'.

The format of the directive is:

```
AddClient .extension user-agent
```

.extension The file extension of the file you want to send to a particular browser.

You cannot use any wildcard characters in this pattern. This extension can be one of a string of suffixes used to qualify a file. For example, the extension .Mozilla can apply to a file named TxtSample.UK.Mozilla.html or TxtSample.html.Mozilla.eng.

user-agent The value to match in the User-Agent header of the incoming request.

This field is case-sensitive. You can use an asterisk (*) as a wildcard character in this field. For example, IBM* applies to all versions of IBM's browser, WebExplorer.

Examples:

```
AddClient .Mozilla      Mozilla/2.*
AddClient .OldMozilla   Mozilla/1.*
AddClient .Webex        IBM*
```

Program default setting: If a client's User-Agent is not matched in an AddClient directive, the server looks for a generic file extension (.htm or .html) to send. If the server cannot find a generic file extension, it uses an algorithm to calculate the quality of all the extensions for that file and sends the file whose extension yields the highest quality, considering it to be the 'best match'.

Initial configuration file setting: None

Using automatic browser detection for Welcome pages

Using automatic browser detection for sending various welcome pages requires additional considerations. Follow these steps to enable browser detection and multi-format processing for your welcome pages. As an example, they show how to serve different versions of the index.html file (in a fictitious webhome directory) but this also works for Welcome.html, welcome.html, or Frontpage.html files.

1. Add a Welcome directive to your configuration file (httpd.cnf or httpd.conf) that specifies the file name with the .multi extension.

Resource mapping

```
Welcome index.multi
```

2. Include AddClient directives in the configuration file that specify which file extensions to send to a particular browser.

```
AddClient .Mozilla      Mozilla/2.*
AddClient .OldMozilla   Mozilla/1.*
AddClient .Webex        IBM*
```

3. Create specific versions of the index file for each of these file extensions: index.Mozilla.html, index.OldMozilla.html, index.WebEx.html.
4. Create a dummy file in the directory called index.multi.
5. Specify only the directory name in the URL when linking to this page.

```
http://www.web4hire.com/webhome/
```

Resource mapping - Redirect URLs

Use the directives described in this section to control which requests your server accepts and where the server looks for resources.

Use the mapping directives (Exec, Fail, Map, Pass, and Redirect) to control which requests your server accepts and to map URL requests to your actual files.

You can use the mapping directives to create a virtual hierarchy of Web resources. You can then change the physical location of files or directories without affecting the virtual layout. Even if your server sends documents from different file systems, it can present a virtual layout.

The server applies the mapping directives in the order they appear in the configuration file until a request has been accepted, rejected, or there are no more directives that apply to the request.

Exec - Run a CGI program for matching requests

Use this directive to specify a template for requests you want to accept and respond to by running a CGI program. Once a request matches a template on an Exec directive, the request is not compared to request templates on any subsequent directives.

The format of the directive is:

```
Exec request-template program-path [Server-IP-address or hostname]
```

request-template

A template for requests that you want your server to accept and respond to by running a CGI program.

Resource mapping

You must use an asterisk as a wildcard in both the *request-template* and *program-path*. The part of the request that matches the *request-template* wildcard must begin with the name of the file that contains the CGI program.

The request can also contain additional data that is passed to the CGI program in the PATH_INFO environment variable. The additional data follows the first slash character that comes after the CGI program file name on the request. The data is passed according to CGI specifications.

program-path

The path to the file that contains the CGI program you want the server to execute for the request. *program-path* must also contain a wildcard. The wildcard is replaced with the name of the file that contains the CGI program.

Important migration note: The Exec directive is now recursive and applies to all subdirectories. You no longer need a separate Exec directive for each directory under cgi-bin and admin-bin.

Server-IP-address or hostname

If you are using multiple IP addresses or virtual hosts, use this parameter to specify an IP address or a host name. (For more information on using multiple IP addresses or virtual hosts, see “Running your server with multiple IP addresses or virtual hosts” on page 19.) The server uses the directive only for requests that come to the server on this IP address or for this host. For an IP address, this is the address of the server’s network connection, not the address of the requesting client.

You can specify an IP address (for example, 204.146.167.72) or you can specify a host name (for example, hostA.bcd.com).

This parameter is optional. Without this parameter, the server uses the directive for all requests regardless of the IP address the requests come in on or the host name in the URL.

Important migration note: Beginning with Version 4.2, a wildcard character can no longer be specified for a server’s IP address.

Examples:

```
Exec /idd/depts/* d:\depts\bin\*
```

In the above example, if your server receives a request of /idd/depts/plan/c92, it runs the CGI program in d:\depts\bin\plan.exe or d:\depts\bin\plan.cmd with c92 passed to the program as input.

```
Exec /cgi-bin/* C:\CGI-BIN\customerA\* 204.146.167.72
Exec /cgi-bin/* C:\CGI-BIN\customerB\* 9.83.100.45
```

The above example uses the optional IP address parameter. If your server receives requests that begin with /cgi-bin/, it serves the request from a different directory based on the IP address of the network connection the request comes in on. For requests coming in on 9.67.106.79, the server uses the C:\CGI-BIN\customerA directory. For requests coming in on any connection with an address of 9.83.100.45, the server uses the C:\CGI-BIN\customerB directory.

Resource mapping

```
Exec    /cgi-bin/*    C:\CGI-BIN\customerA\*    hostA.bcd.com
Exec    /cgi-bin/*    C:\CGI-BIN\customerB\*    hostB.bcd.com
```

The above example uses the optional host name parameter. If your server receives requests that begin with /cgi-bin/, it serves the request from a different directory based on the host name in the URL. For requests coming in for hostA.bcd.com, the server uses the C:\CGI-BIN\customerA directory. For requests coming in on any connection for hostB.bcd.com, the server uses the C:\CGI-BIN\customerB directory.

Initial configuration file setting

```
Exec    /admin-bin/*    d:\path\*
Exec    /cgi-bin/*     d:\path\*
```

d:\path is replaced by a directory you entered at installation. If you used the installation defaults, your default Exec directives would be:

```
Exec    /admin-bin/*    C:\WWW\ADMIN\*
Exec    /cgi-bin/*     C:\WWW\CGI-BIN\*
```

Fail - Reject matching requests

Use this directive to specify a template for requests you do not want to process. Once a request matches a template on a Fail directive, the request is not compared to request templates on any subsequent directives.

The format of the directive is:

```
Fail request-template [Server-IP-address or hostname]
```

request-template

A template for requests that you want your server to reject. If a request matches the template, the server sends the requester an error message.

You can use an asterisk as a wildcard in the template. The tilde character just after a slash (/) has to be explicitly matched; a wildcard cannot be used to match it.

Server-IP-address or *hostname*

If you are using multiple IP addresses or virtual hosts, use this parameter to specify an IP address or a host name. (For more information on using multiple IP addresses or virtual hosts, see “Running your server with multiple IP addresses or virtual hosts” on page 19.) The server uses the directive only for requests that come to the server on this IP address or for this host. For an IP address, this is the address of the server’s network connection, not the address of the requesting client.

You can specify an IP address (for example, 204.146.167.72) or you can specify a host name (for example, hostA.bcd.com).

Resource mapping

This parameter is optional. Without this parameter, the server uses the directive for all requests regardless of the IP address the requests come in on or the host name in the URL.

Important migration note: Beginning with Version 4.2, a wildcard character can no longer be specified for a server's IP address.

Examples:

```
Fail /usr/local/private/*
```

In the above example, the server rejects any requests beginning with /usr/local/private/.

```
Fail /customerB/* 204.146.167.72
Fail /customerA/* 9.83.100.45
```

The above example uses the optional IP address parameter. The server rejects any requests beginning with /customerB/ if the request comes in on a network connection with IP address 204.146.167.72. The server rejects any requests beginning with /customerA/ if the request comes in on a network connection with an IP address of 9.83.100.45.

```
Fail /customerB/* hostA.bcd.com
Fail /customerA/* hostB.bcd.com
```

The above example uses the optional host name parameter. The server rejects any requests beginning with /customerB/ if the request comes in for hostA. The server rejects any requests beginning with /customerA/ if the request comes in for hostB.

Initial configuration file setting: None.

Map - Change matching requests to a new result string

Use this directive to specify a template for requests you want to change to a new request string. After your server changes the request, it takes the new request string and compares it to the request templates on subsequent directives.

The format of the directive is:

```
Map request-template new-request [Server-IP-address or hostname]
```

request-template

A template for requests that you want your server to change and then continue comparing the new request string to other templates.

You can use an asterisk as a wildcard in the template. The tilde character just after a slash (/) has to be explicitly matched; a wildcard cannot be used to match it.

Resource mapping

new-request

The new request string you want your server to continue to compare to the request templates on subsequent directives. *new-request* may contain a wildcard if the *request-template* has one. The part of the request that matches the *request-template* wildcard is inserted in place of the wildcard in *new-request*.

Server-IP-address or *hostname*

If you are using multiple IP addresses or virtual hosts, use this parameter to specify an IP address or a host name. (For more information on using multiple IP addresses or virtual hosts, see “Running your server with multiple IP addresses or virtual hosts” on page 19.) The server uses the directive only for requests that come to the server on this IP address or for this host. For an IP address, this is the address of the server’s network connection, not the address of the requesting client.

You can specify an IP address (for example, 204.146.167.72) or you can specify a host name (for example, hostA.raleigh.ibm.com).

This parameter is optional. Without this parameter, the server uses the directive for all requests regardless of the IP address the requests come in on or the host name in the URL.

Important migration note: Beginning with Version 4.2, a wildcard character can no longer be specified for a server’s IP address.

Examples:

```
Map /stuff/* /good/stuff/*
```

In the above example, your server would take any requests starting with `/stuff/` and change the `/stuff/` portion of the request to `/good/stuff/`. Anything that followed `/stuff/` on the original request would also be included in the new request string. So `/stuff/whatsup/` would change to `/good/stuff/whatsup/`. Your server would take the new request string and continue to compare it to request templates on subsequent directives.

```
Map /stuff/* /customerA/good/stuff/* 204.146.167.72
Map /stuff/* /customerB/good/stuff/* 9.83.104.45
```

The above examples use the optional IP address parameter. If your server receives requests that begin with `/stuff/`, it changes the request to a different request string based on the IP address of the network connection the request comes in on. For requests coming in on 204.146.167.72 the server changes the `/stuff/` portion of the request to `/customerA/good/stuff/`. For requests coming in on any connection with an address of 9.83.100.45, the server changes the `/stuff/` portion of the request to `/customerB/good/stuff/`.

```
Map /stuff/* /customerA/good/stuff/* hostA.bcd.com
Map /stuff/* /customerB/good/stuff/* hostB.bcd.com
```

The above examples use the optional host name parameter. If your server receives requests that begin with `/stuff/`, it changes the request to a different request string

Resource mapping

based on the host name in the URL. For requests coming in for hostA, the server changes the /stuff/ portion of the request to /customerA/good/stuff/. For requests coming in for hostB, the server changes the /stuff/ portion of the request to /customerB/good/stuff/.

Initial configuration file setting None.

Pass - Accept matching requests

Use this directive to specify a template for requests you want to accept and respond to with a document from your server. Once a request matches a template on a Pass directive, the request is not compared to request templates on any subsequent directives.

The format of the directive is:

```
Pass request-template [file-path [Server-IP-address or hostname]]
```

request-template

A template for requests that you want your server to accept and respond to with a document from your server.

You can use an asterisk as a wildcard in the template. The tilde character just after a slash (/) has to be explicitly matched; a wildcard cannot be used to match it.

file-path

The path to the file that contains the document you want the server to return. *file-path* may contain a wildcard if the *request-template* has one. The part of the request that matches the *request-template* wildcard is inserted in place of the wildcard in *file-path*.

This parameter is optional. If you do not specify a path, the request itself is used as the path. The drive is assumed to be the drive where the server program is installed.

Server-IP-address or hostname

If you are using multiple IP addresses or virtual hosts, use this parameter to specify an IP address or a host name. (For more information on using multiple IP addresses or virtual hosts, see "Running your server with multiple IP addresses or virtual hosts" on page 19.) The server uses the directive only for requests that come to the server on this IP address or for this host. For an IP address, this is the address of the server's network connection, not the address of the requesting client.

You can specify an IP address (for example, 204.146.167.72) or you can specify a host name (for example, hostA.raleigh.ibm.com).

This parameter is optional. Without this parameter, the server uses the directive for all requests regardless of the IP address the requests come in on or the host name in the URL.

Resource mapping

Important migration note: Beginning with Version 4.2, a wildcard character can no longer be specified for a server's IP address.

Examples:

```
Pass /updates/parts/* C:\WWW\HTML\catalog\updates\parts\*
```

In the above example, your server would respond to a request starting /updates/parts/ with a document from C:\WWW\HTML\catalog\updates\parts\. Anything that followed /updates/parts/ would also be used to identify the document. So your server would respond to the request /updates/parts/printers/ribbon.html with the document in file C:\WWW\HTML\catalog\updates\parts\printers\ribbon.html.

```
Pass /gooddoc/*
```

In the above example, your server would respond to a request starting with /gooddoc/ with a document from *d*:\gooddoc\ (where *d*: is the drive where you installed the server program). So your server would respond to the request /gooddoc/volume1/issue2/newsletter4.html with the document in file *d*:\gooddoc\volume1\issue2\newsletter4.html.

```
Pass /parts/* C:\customerA\catalog\* 204.146.167.72
Pass /parts/* C:\customerB\catalog\* 9.83.104.45
```

The above examples use the optional IP address parameter. If your server receives requests that begin with /parts/, it returns a file from a different directory based on the IP address of the network connection the request comes in on. For requests coming in on 204.146.167.72 the server returns a file from C:\customerA\catalog\. For requests coming in on any connection with an address of 9.83.104.45, the server returns a file from C:\customerB\catalog\.

```
Pass /parts/* C:\customerA\catalog\* hostA.bcd.com
Pass /parts/* C:\customerB\catalog\* hostB.bcd.com
```

The above examples use the optional host name parameter. If your server receives requests that begin with /parts/, it returns a file from a different directory based on the IP address of the network connection the request comes in on. For requests coming in for hostA.bcd.com, the server returns a file from C:\customerA\catalog\. For requests coming in for hostB.bcd.com, the server returns a file from C:\customerB\catalog\.

Initial configuration file setting

```
Pass /reports/* d:\path\*
Pass /Docs/* d:\path\*
Pass /httpd-internal-icons/* d:\path\*
Pass /icons/* d:\path\*
Pass /Admin/*.html d:\path\*
Pass /Admin/*.gif d:\path\*
Pass /* d:\path\*
```

d:\path is replaced by a directory you entered at installation. If you used the installation defaults, these Pass directives would be:

Resource mapping

| | | |
|------|-------------------------|----------------|
| Pass | /Docs/* | C:\WWW\DOCS* |
| Pass | /httpd-internal-icons/* | C:\WWW\ICONS* |
| Pass | /icons/* | C:\WWW\ICONS* |
| Pass | /Admin/* | C:\WWW\ADMIN* |
| Pass | /* | C:\WWW\HTML* |

Note: The path following Pass /* is your document root directory. See “Understanding the document root directory” on page 8 for more information.

Redirect - Send matching requests to another server

Use this directive to specify a template for requests you want to accept and send to another server. Once a request matches a template on a Redirect directive, the request is not compared to templates on any other directives in your configuration file.

The format of the directive is:

```
Redirect request-template URL [Server-IP-address or hostname]
```

request-template

A template for requests that you want your server to send to another server.

You can use an asterisk as a wildcard in the template. The tilde character just after a slash (/) has to be explicitly matched; a wildcard cannot be used to match it.

URL

The URL request you want your server to send to another server. The response to this request goes to the original requester without any indication that it did not come from your server.

URL must contain a protocol specification and the name of the server to send the request to. It can also contain a path or file name. If *request-template* uses a wildcard, the path or file name on *URL* can also use a wildcard. The part of the original request that matches the wildcard on *request-template* is inserted in place of the wildcard on *URL*.

Server-IP-address or *hostname*

If you are using multiple IP addresses or virtual hosts, use this parameter to specify an IP address or a host name. (For more information on using multiple IP addresses or virtual hosts, see “Running your server with multiple IP addresses or virtual hosts” on page 19.) The server uses the directive only for requests that come to the server on this IP address or for this host. For an IP address, this is the address of the server’s network connection, not the address of the requesting client.

You can specify an IP address (for example, 204.146.167.72) or you can specify a host name (for example, hostA.bcd.com).

Resource mapping

This parameter is optional. Without this parameter, the server uses the directive for all requests regardless of the IP address the requests come in on or the host name in the URL.

Important migration note: Beginning with Version 4.2, a wildcard character can no longer be specified for a server's IP address.

Example

```
Redirect /chief/stuff/* http://www.other.org/wahoo/*
```

In this example, your server sends any requests beginning with /chief/stuff/ to the wahoo directory of the www.other.org server.

```
Redirect /stuff/* http://www.chief.org/wahoo/* 204.146.167.72
Redirect /stuff/* http://www.dawg.com/pound/* 9.83.100.45
```

The above examples use the optional IP address parameter. If your server receives requests that begin with /stuff/, it redirects the request to different servers based on the IP address of the network connection the request comes in on. For requests coming in on 204.146.167.72, the server sends the request to the wahoo directory of the www.chief.org server. For requests coming in on any connection with an address of 9.83.100.45, the server sends the request to the pound directory of the www.dawg.com server.

```
Redirect /stuff/* http://www.chief.org/wahoo/* hostA.bcd.com
Redirect /stuff/* http://www.dawg.com/pound/* hostB.bcd.com
```

The above examples use the optional IP address parameter. If your server receives requests that begin with /stuff/, it redirects the request to different servers based on the host name in the URL. For requests coming in for hostA, the server sends the request to the wahoo directory of the www.chief.org server. For requests coming in for hostB, the server sends the request to the pound directory of the www.dawg.com server.

Initial configuration file setting: None.

InheritEnv - Specify which environment variables are inherited by CGI programs

Use this directive to specify which environment variables you want your CGI programs to inherit (other than the CGI environment variables that are specific to CGI processing).

If you do not include an InheritEnv directive, all environment variables are inherited by CGI programs. If you include any InheritEnv directive, only those environment variables specified on InheritEnv directives will be inherited along with the CGI-specific environment variables. The directive allows you to optionally initialize the value of the variables that are inherited.

Resource mapping

Refer to the *Web Programming Guide* for a list of the CGI-specific environment variables.

Example

```
InheritEnv PATH
InheritEnv LANG=ENUS
```

In this example, only the PATH and LANG environment variables will be inherited by CGI programs.

Initial configuration file setting: None. The default is all environment variables are inherited by CGI programs.

DisInheritEnv - Specify which environment variables are disinherited by CGI programs

Use this directive to specify which environment variables you do not want your CGI programs to inherit (other than the CGI environment variables that are specific to CGI processing).

By default, all environment variables are inherited by CGI programs. You can exclude individual environment variables from being inherited with the DisInheritEnv directive.

Refer to the *Web Programming Guide* for a list of the CGI-specific environment variables.

Example

```
DisInheritEnv PATH
DisInheritEnv LANG
```

In this example, all environment variables except PATH and LANG will be inherited by CGI programs.

Initial configuration file setting: None. The default is all environment variables are inherited by CGI programs.

Error message customization - Customize error messages the server returns to clients

Use this directive to customize the messages your server sends to the requesting client when it encounters an error condition. For example, you can change a message to include more information about the cause of the problem and suggest possible solutions to fix it. For internal networks, you might provide a contact person for your users to call.

Each error condition is identified by a key word. To decide which error messages you want to customize, first review the list of error conditions, their causes, and the default message that the server sends. Then, for each error message you want to change:

- Create an individual HTML file with the desired text.
- Add an **ErrorPage** directive to your configuration file that associates the error condition key word with the HTML file you want to serve.

Note: The server does not parse your error files for imbeds, regardless of the file extensions or use of the Imbeds directive.

ErrorPage - Specify a customized message for a particular error condition

Use this directive to specify the name of a file that you want to send when the server encounters a particular error condition.

You can place this directive anywhere in the configuration file. When the error occurs, the file will be processed according to the mapping rules defined in your configuration file. Therefore, the file you want to send must be in a location that can be reached through the mapping rules as defined by the Fail, Map, NameTrans, Pass, Redirect, Service directives. At a minimum, you need a Pass directive that would allow the server to pass the error message file.

The format of this directive is:

```
ErrorPage keyword /path/filename.htm
```

| | |
|---------------------------|---|
| <i>keyword</i> | One of the key words associated with an error condition. See “Error Conditions, Causes, and Default Messages” on page 112 for a list of keywords. |
| <i>/path/filename.htm</i> | This is the fully qualified Web name of your error file, as viewed by a client on the Web. The file must be an HTML file. |

Example

Error message customization

```
ErrorPage scriptstart /errors/html/scriptstart.htm
```

In the above example, when a **scriptstart** condition is encountered, the server will send the scriptstart.htm file to the client.

This file might contain the following HTML text:

```
<HTML>
<HEAD>
<TITLE>Message for SCRIPTSTART condition</TITLE>
</HEAD>
<BODY>
The CGI program could not be started.
<P>
<A HREF="mailto:admin@websvr.com">Notify the administrator</A>
of this problem.
</BODY>
</HTML>
```

If the server's configuration file contains `PASS /* d:\wwwhome*`, then the full path for this message file would be `d:\wwwhome\errors\html\scriptstart.htm`.

Default:: If you do not specify an **ErrorPage** directive for an error condition, the server's default error message for that condition will be sent.

Error Conditions, Causes, and Default Messages

The following list shows the HTTP response code and key word for each error condition, followed by the probable cause, and the default message the server sends.

| Code and Key Word | Cause and Default Message |
|--------------------------|---|
| 302 okredirect | <p>Cause: The requested file is on another recognized server. The name of the server is sent back to the requesting client along with a message. The client can connect to the correct server or display the message that is sent.</p> <p>Default message: Found.</p> |
| 400 badrequest | <p>Cause: Either there is a network problem, such as a time-out, or the request was indecipherable.</p> <p>Default message: Invalid request - completely unable to parse it.</p> |
| 400 badscript | <p>Cause: The server could determine that the requested file was a CGI script but it could not process it; the request was invalid in some way.</p> |

Error message customization

| | |
|--------------------------|--|
| | <p>Default message: The script execution request is not valid.</p> |
| 400 connectfailed | <p>Cause: On a tunneled request, the server could not connect to the requested partner on the requested port.</p> <p>Default message: Host not found or not responding.</p> |
| 400 nopartner | <p>Cause: On a tunneled request, the server could not connect to the requested hostname due to bad syntax or an unknown host.</p> <p>Default message: Host not found or not responding.</p> |
| 400 proxyfail | <p>Cause: The client is trying to use the server as a proxy, and although this is allowed, it did not work. Possibly the destination server doesn't exist or is busy.</p> <p>Default message: Proxy load failed.</p> |
| 400 unknownmethod | <p>Cause: The request did not include a recognized method, such as GET, POST, PUT, or DELETE.</p> <p>Default message: The request is not valid or not recognized.</p> |
| 401 notauthorized | <p>Cause: The request requires a user ID and password. Either the user ID and password sent by the client are not valid for this request or the client did not send a user ID and password.</p> <p>Default message: Not Authorized. Authentication failed.</p> |
| 401 notmember | <p>Cause: The requested file has a protection rule listing valid user IDs and passwords and the user ID of the requesting client is not included in that list.</p> <p>Default message: Not authorized to access the document.</p> |
| 403 baduser | <p>Cause: The client requested a user's home directory that does not exist.</p> <p>Default message: The user directory is not valid.</p> |
| 403 badredirect | <p>Cause: The server is trying to redirect the request and the Redirect directive is invalid (possibly missing a destination) or contains a loop.</p> <p>Default message: The redirection in the configuration file is not valid.</p> |
| 403 byrule | <p>Cause: Either the file requested is specifically blocked by a Fail directive or it does not match any of the files that are allowed to be accessed according to other request mapping directives.</p> <p>Default message: Forbidden by rule.</p> |

Error message customization

| | |
|---------------------------|--|
| 403 dirbrowse | <p>Cause: The client specified a directory (rather than a file name) in the URL that does not have a welcome page and the administrator has turned off directory browsing (either for this directory or for the entire server).</p> <p>Default message: Directory browsing failed - access forbidden.</p> |
| 403 dotdot | <p>Cause: The client request contains an instruction (<code>/../</code>) to navigate above the document directory root and this is not allowed.</p> <p>Default message: Forbidden - URL containing <code>..</code> forbidden (don't try to break in).</p> |
| 403 ipmask | <p>Cause: The file requested has a protection rule that includes a list of valid IP addresses and the client's address is not included in the list.</p> <p>Default message: Server will not serve to your IP address.</p> |
| 403 ipmaskproxy | <p>Cause: The client is trying to use the server as a proxy and the client is not included in the list of host names or IP addresses that are allowed to do so.</p> <p>Default message: Proxy server will not serve to your IP address (at least with this HTTP method).</p> |
| 403 methoddisabled | <p>Cause: The client requested a method (such as GET, POST, PUT, DELETE) that is specifically not allowed by the Disable directive.</p> <p>Default message: Method <i>method</i> is disabled on this server.</p> |
| 403 noacl | <p>Cause: The directory has a protection rule but does not have an Access Control List (ACL) defined and the protection setup does not have a GetMask subdirective. The administrator needs to remove the protection rule or add an ACL.</p> <p>Default message: Access to this file is not allowed 'no ACL file'.</p> |
| 403 noentry | <p>Cause: The directory is protected by an Access Control List (ACL) and the user is not included in the ACL.</p> <p>Default message: Access to this file is not allowed (no ACL entry).</p> |
| 403 notallowed | <p>Cause: The requested file was found but the server's protection setup prevented access. This is commonly generated for URLs that point to CGI programs.</p> <p>Default message: The PUT and DELETE methods must be specified in the server's protection setup.</p> |

Error message customization

| | |
|-----------------------------|---|
| 403 openfailed | <p>Cause: After passing the protection rules, the server determined that the client should have read access to the file but the operating system will not allow the server to access it. Possibly the user ID running the server does not have read permission to the file it is trying to serve or the file system may be encountering problems.</p> <p>Default message: Can't browse selected file.</p> |
| 403 setuperror | <p>Cause: The directory has an Access Control List (ACL) defined but does not have a protection rule. The administrator needs to add a protection rule or remove the ACL.</p> <p>Default message: Server protection setup error occurred. Probably, the protection setup file was not found or it contained a syntax error.</p> |
| 404 multifail | <p>Cause: The requested file could not be found on the server. The server tried to match the file name exactly as specified and with every known file extension appended.</p> <p>Default message: The file was not found, even after searching on any extensions to the file name.</p> |
| 407 proxynotauth | <p>Cause: The proxy request requires a user ID and password. Either the user ID and password sent by the client are not valid for this request or the client did not send a user ID and password. Note that some Web browsers do not support the PROXY-AUTHENTICATE function.</p> <p>Default message: Not authorized. Proxy-Authentication failed (or your browser does not support it).</p> |
| 407 proxynotmember | <p>Cause: The proxy request has a protection rule listing valid user IDs and the user ID of the requesting client is not included in that list.</p> <p>Default message: Not authorized for proxy access to the document.</p> |
| 412 preconditionfail | <p>Cause: A precondition specified by the client on this request was not met. For example, this could result from an HTTP/1.1 request with a condition "if-not-modified-since xxx".</p> <p>Default message: Precondition failed: could not match entity tags.</p> |
| 500 scriptio | <p>Cause: The client requested a CGI script; the server can find it and start it but cannot get it to process input or output. The script may contain invalid code.</p> <p>Default message: Cannot read script output pipe.</p> |

Timeouts

| | |
|---------------------------|---|
| 500 scriptnotfound | <p>Cause: The client requested a CGI script that cannot be found.</p> <p>Default message: The script request is not valid; none of <i><program></i> and <i><program>.pp</i> is executable.</p> |
| 500 scriptstart | <p>Cause: The client requested a CGI script; the server can find it but cannot start it. The script may contain invalid code.</p> <p>Default message: Starting the CGI program failed. Could not communicate with the CGI program.</p> |
| 501 noformat | <p>Cause: The server has encountered an internal error and cannot interpret the format of the file it is trying to serve. The file may be corrupted or have an unknown or invalid file extension.</p> <p>Default message: Sorry, can't convert from <i>mime-type-1</i> to <i>mime-type-2</i>.</p> |

Timeouts - Close connections automatically

Use the directives described in this section to control the amount of time the server spends processing requests. If you are using persistent connections, see "PersistTimeout - Specify time to wait for the client to send another request" on page 148.

InputTimeout - Specify input timeout setting

Use this directive to set the time allowed for a client to send a request after making a connection to the server. A client first connects to the server and then sends a request. If the client does not send a request within the amount of time on this directive, the server drops the connection. Specify the time value in any combination of hours, minutes (or mins), and seconds (or secs).

Example

```
InputTimeout 3 mins 30 secs
```

Initial configuration file setting

```
InputTimeout 5 minutes 30 seconds
```

Program default setting

```
InputTimeout 2 minutes
```

OutputTimeout - Specify output timeout setting

Use this directive to set the maximum time allowed for your server to send output to a client. The time limit applies to requests for local files and requests for which the server is acting as a proxy. The time limit does not apply for requests that start a local CGI program.

If the server does not send the complete response within the amount of time on this directive, the server drops the connection. Specify the time value in any combination of hours, minutes (or mins), and seconds (or secs).

Example

```
OutputTimeout 20 minutes
```

Initial configuration file setting

```
OutputTimeout 1 hour
```

Program default setting

```
OutputTimeout 20 minutes
```

ScriptTimeout - Specify script timeout setting

Use this directive to set the time allowed for a program started by the server to finish. The server stops a program if it runs longer than the limit. Specify the time value in any combination of hours, minutes (or mins), and seconds (or secs).

Example

```
ScriptTimeout 15 mins
```

Initial configuration file setting

```
ScriptTimeout 10 minutes
```

Program default setting

```
ScriptTimeout 5 minutes
```

Methods

Methods - Set method acceptance

Use the directives described in this section to control which HTTP methods are enabled for your server.

Client requests to the server include a method field that indicates the action the server is to perform on the specified object. The request identifies the object with a Uniform Resource Locator (URL).

Following is a list of methods that the server supports and a description of how the server would respond to a client request containing the method. The description assumes the method is enabled.

- **CONNECT**- This method is used to establish an SSL tunneling session between a client, such as NetScape Navigator, and a remote server through a proxy server. The sessions between the client and the proxy and between the proxy and the remote server are secure. The proxy does not have access to the data being sent. The proxy server can be a base or secure server.
- **DELETE** - The server deletes the object identified by the URL. After the object is deleted, the URL is not valid. Because delete typically lets clients delete information from your server, you must use protection setups to define who can use this method and which files can be deleted (see Chapter 7, "Protecting your server" on page 201).
- **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.
- **HEAD** - The server returns only HTTP document headers without the document body.
- **OPTIONS** - The request returns information about the communications options on the request/response chain identified by the URL. This method allows a client to determine the options and requirements associated with an object, or the capabilities of a server, without having to act on or retrieve the object.
- **POST** - The request contains data and a URL. The server accepts the data enclosed in the request as a new subordinate of the resource identified in the URL. The resource, which may be a data-accepting program, a gateway to some other protocol, or a separate program that accepts annotations, processes the enclosed data. The POST method is designed to handle annotation of existing resources; posting of a message to a bulleting boards, newsgroup, mailing list, or similar group of articles; providing a block of data, such as data from a form to a data-handling program; or extending a database through an append operation. In the Internet Connection Secure Server, the POST method is used to process the Configuration and Administration forms.
- **PUT** - The request contains data and a URL. The server stores the resource identified in the URL. If the resource already exists, PUT replaces it. If the resource does not exist, PUT creates it. Because PUT typically lets clients add or

Methods

replace information on your server, you must use protection setups to define who can use this method for which files (see Chapter 7, “Protecting your server” on page 201).

- TRACE - The server echos the request message sent by the client. This method allows the client to see what is being received at the other end of the request chain and use that data for testing or diagnostic information. The content type of the response is message/http.

Disable - Disable HTTP methods

Use this directive to specify which HTTP methods you do not want your server to accept.

In the default configuration file, the GET, HEAD, OPTIONS, POST, and TRACE methods are enabled and all other supported HTTP methods are disabled. To disable a method that is currently enabled, change the Enable directive for the method to a Disable directive.

Note: The Configuration and Administration forms use the POST method to make updates to your server configuration. If you disable the POST method you will not be able to use the Configuration and Administration forms.

Example

```
Disable HEAD
```

Initial configuration file setting

```
Disable PUT
Disable DELETE
```

Enable - Enable HTTP methods

Use this directive to specify which HTTP methods you want your server to accept.

You can enable as many of the HTTP methods as you need. For each method you want the server to accept, enter a separate Enable directive followed by the name of the method.

Example

```
Enable DELETE
```

If no Service directive exists for a particular URL, you can use the Enable directive to perform customized programming for any HTTP method. The program you specify on this directive will override the standard processing for that method.

The format is:

Meta-Information

```
Enable    method d:\path\fileDLL:function_name
```

Example

```
Enable
```

Initial configuration file setting

```
Enable GET
Enable HEAD
Enable POST
Enable TRACE
Enable OPTIONS
```

Important migration notes

- Beginning with Version 4.2, the CHECKIN and CHECKOUT methods are no longer supported. You can use the PUT method for CHECKIN and the GET method for CHECKOUT; however, the Internet Connection Secure Server does not perform locking or versioning.
- The accessory script directives, DELETE-Script, POST-Script, PUT-Script, and Search, are no longer supported. You need to port your scripts to ICAP1 applications. See the *Web Programming Guide* for more information.

Meta-Information - Name meta-information files and directories

Use the directives described in this section to control where your server looks for meta-information files.

You can use a separate set of files to store meta-information about your server's documents. The server can include the meta-information with its HTTP responses. Meta-information describes the file containing a document, not the contents of the document. For example, meta-information for a file might give the date the file was created and the date it was last modified. You can include any valid response headers as described in the HTTP 1.1 specification.

HTTP recognizes MIME headers. Information that MIME header fields can include are the file type, subtype, encoding, and content length.

Each line of a meta-information file contains a header field, followed by a colon, and the value of the field. For example:

```
Last-Modified: Wednesday, 05-Apr-96 20:51:35 GMT
Expires: Friday, 30-Jun-96 24:00:00 GMT
MIME-Version: 1.0
```

MetaDir - Specify name of subdirectory for meta-information files

Use this directive to specify the name you want to use for subdirectories that contain meta-information files. You can only have one instance of this directive, which means all your meta-information subdirectories have the same name.

Any directory from which your server retrieves files can have a subdirectory with the name specified on this directive. The files on the meta-information subdirectory contain meta-information about the files being retrieved. The meta-information files have the same file name and extension as the file they describe, plus an added extension. The name of the added extension is specified on the MetaSuffix directive.

For example, you might have the following two directives in your configuration file:

```
MetaDir    look_here
MetaSuffix .file_desc
```

If your server goes to retrieve this file:

```
d:/html/realcool/coolindex.html
```

it looks for meta information to include with the response in this file:

```
d:/html/realcool/look_here/coolindex.html.file_desc
```

Example

```
MetaDir mimeinfo
```

Initial configuration file setting

```
MetaDir .web
```

Note: The dot character (.) at the beginning of the default value is used as part of the subdirectory name.

MetaSuffix - Specify the extension for meta-information files

Use this directive to specify the extension you want to use for meta-information files. You can only have one instance of this directive, which means all meta-information files end with the same extension. You must include the period character (.) as part of the value.

Any file your server retrieves can have a meta-information file associated with it. A meta-information file has the same file name and extension as the file it describes, plus the additional extension specified on the MetaSuffix directive. A meta-information file must be located on a subdirectory of the directory that contains the file being described. The name of the subdirectory must be the name specified on the MetaDir directive.

ICAPI application processing

See the description of the MetaDir directive to see an example of how MetaDir and MetaSuffix work together.

Example

```
MetaSuffix .head
```

Initial configuration file setting

```
MetaSuffix .meta
```

ICAPI application processing - Specify ICAPI applications for request processing

The Internet Connection Application Programming Interface (ICAPI) allows you to extend the Internet Connection Secure Server's base functions with your own customized processing routines. Use the directives described in this section to have the server call the application functions in your program at various points in its request processing cycle. You can find detailed information for writing the application functions and compiling your program in the *Web Programming Guide*.

Except for Service and NameTrans, these directives can be in any order in the configuration file and you do not need to include every directive. If you do not have a customized application function for a particular step, just omit the corresponding directive. The normal processing for that step will execute by default.

The Service and NameTrans directives behave like the other mapping directives and are sensitive to their placement in the configuration file. For example a rule for `/cgi-bin/foo.dll` must appear before the rule for `/cgi-bin/*`.

You can also have more than one configuration directive for a step. For example, you could include two NameTrans directives, each pointing to a different application function. When the server performs the name translation step, it will process your name translation functions in the order in which they appear within the configuration file.

Your application functions do not have to be executed for every request:

- By specifying a URL with some directives, you can indicate that you want the application function called only for URLs that match a certain pattern or mask.
- By specifying an authentication scheme with the Authentication directive, you can indicate that you want the application function called only for certain types of authentication.

ServerInit - Customize the Server Initialization step

Use this directive to specify a customized application function you want the server to call during its initialization routines. This code will be executed before any client requests are read and whenever the server is restarted.

If you are using the GoServe modules in the PreExit or Service steps, you need to call the gosclone module here.

The format of the directive is:

```
ServerInit d:\path\file:function_name
```

d:\path\file

The fully qualified file name of your compiled program, including the extension.

:function_name

The name you gave your application function within your program.

Examples:

```
ServerInit d:\ics\api\bin\icsext05.dll:svr_init
```

```
ServerInit d:\www\dll\gosclone.dll:init
```

Initial configuration file setting: None.

PreExit - Customize the PreExit step

Use this directive to specify a customized application function you want the server to call during the PreExit step. This code will be executed after a client request has been read but before any other processing occurs. You can call the GoServe module during this step.

The format of the directive is:

```
PreExit d:\path\file:function_name
```

d:\path\file

The fully qualified file name of your compiled DLL, including the extension.

:function_name

The name you gave your application function within your program.

Examples:

```
PreExit d:\ics\api\bin\icsext05.dll:pre_exit
```

```
PreExit d:\www\dll\gosclone.dll:goserve
```

ICAPI application processing

Initial configuration file setting: None.

Authentication - Customize the Authentication step

Use this directive to specify a customized application function you want the server to call during the Authentication step. This code will be executed based on the authentication scheme. Currently, only **Basic** authentication is supported.

Note: Authentication is part of the authorization process; it only occurs when authorization is required.

The format of the directive is:

```
Authentication type d:\path\file:function_name
```

type

Specifies an authentication scheme which further determine if your application function is called. Both an asterisk (*) and Basic are accepted values.

d:\path\file

The fully qualified file name of your compiled program, including the extension.

:function_name

The name you gave your application function within your program.

Example

```
Authentication BASIC d:\ics\api\bin\icsextpgm.dll:basic_authen
```

Initial configuration file setting: None.

NameTrans - Customize the Name Translation step

Use this directive to specify a customized application function you want the server to call during the Name Translation step. This code would supply the mechanism for translating the virtual path in the request to the physical path on the server, mapping URLs to specific objects.

Note: This is not a terminal mapping rule. The transformed URL still has to match one of the terminal mapping rule directives, such as Exec, Fail, Map, Pass, Redirect, and Service.

The format of the directive is:

```
NameTrans request-template d:\path\file:function_name [Server-IP-address or hostname]
```

ICAPI application processing

request-template

A template for requests that further determine if your application function is called. The specification can include the protocol, domain and host, can be preceded by a slash (/), and can use an asterisk (*) as a wildcard. For example, /front_page.html, http://www.ics.raleigh.ibm.com, /pub*, /*, and * are all valid.

d:\path\file

The fully qualified file name of your compiled program, including the extension.

:function_name

The name you gave your application function within your program.

Server-IP-address or hostname

If you are using multiple IP addresses or virtual hosts, determines if your application function will be called only for requests coming in on a specific IP address or for a specific host.

Important migration note: Beginning with Version 4.2, a wildcard character can no longer be specified for a server's IP address.

Example

```
NameTrans /index.html d:\api\bin\icsextpgm.dll:trans_url
```

Initial configuration file setting: None.

Authorization - Customize the Authorization step

Use this directive to specify a customized application function you want the server to call during the Authorization step. This code would verify that the requested object can be served to the client.

The format of the directive is:

```
Authorization request-template d:\path\file:function_name
```

request-template

A template for requests that further determine if your application function is called. The specification can include the protocol, domain and host, can be preceded by a slash (/), and can use an asterisk (*) as a wildcard. For example, /front_page.html, http://www.ics.raleigh.ibm.com, /pub*, /*, and * are all valid.

d:\path\file

The fully qualified file name of your compiled program, including the extension.

:function_name

The name you gave your application function within your program.

Example

```
Authorization /index.html d:\api\bin\icsextpgm.dll:trans_url
```

ICAPI application processing

Initial configuration file setting: None.

ObjectType - Customize the Object Type step

Use this directive to specify a customized application function you want the server to call during the Object Type step. This code would locate the requested object in the file system and identify its MIME type.

The format of the directive is:

```
ObjectType request-template d:\path\file:function_name
```

request-template

A template for requests that further determine if your application function is called. The specification can include the protocol, domain and host, can be preceded by a slash (/), and can use an asterisk (*) as a wildcard. For example, /front_page.html, http://www.ics.raleigh.ibm.com, /pub*, /*, and * are all valid.

d:\path\file

The fully qualified file name of your compiled program, including the extension.

:function_name

The name you gave your application function within your program.

Example

```
ObjectType /index.html d:\api\bin\icsextpgm.dll:obj_type
```

Initial configuration file setting: None.

Service - Customize the Service step

Use this directive to specify a customized application function you want the server to call during the Service step. This code would service the client request. For example, it sends the file or runs the CGI program.

There is no default for this directive. If the request matches a Service rule (an application function specified on a Service directive is executed) but it returns HTTP_NOACTION, the server will generate an error and the request will fail.

The format of the directive is:

```
Service request-template d:\path\file:function_name [Server-IP_address or hostname]
```

request-template

A template for requests that further determine if your application function is called. The specification can include the protocol, domain and host, can be preceded by a slash (/), and can use an asterisk (*) as a wildcard. For example, /front_page.html, http://www.ics.raleigh.ibm.com, /pub*, /*, and * are all valid.

ICAPI application processing

d:\path\file

The fully qualified file name of your compiled program, including the extension.

:function_name

The name you gave your application function within your program.

Server-IP_address or hostname

If you use multiple IP addresses or virtual hosts, determines if your application function will be called only for requests coming in on a specific IP address or for a specific host.

Important migration note: Beginning with Version 4.2, a wildcard character can no longer be specified for a server's IP address.

Note: If you want full path translation, including *query_string*, you must have an asterisk (*) in both the *request-template* and in the *d:\path\file:function_name* as shown in the second example.

Example

```
Service /index.html d:\ics\api\bin\icsext05.dll:serve_req
```

```
Service /cgi-bin/htimage* \ics\api\htimage:Htimage*
```

Initial configuration file setting: None.

PICSDBLookup - Customize the PICS label retrieval step

Use this directive to specify a customized application function you want the server to call to retrieve PICS labels for a specified URL. Your function can either dynamically create a PICS label for the requested document or to search for a PICS label in an alternative file or database.

The format of the directive is:

```
PICSDBLookup request-template d:\path\file:function_name
```

request-template

A template for requests that further determine if your application function is called. The specification can include the protocol, domain and host, can be preceded by a slash (/), and can use an asterisk (*) as a wildcard. For example, /front_page.html, http://www.ics.raleigh.ibm.com, /pub*, /*, and * are all valid.

d:\path\file

The fully qualified file name of your compiled program, including the extension.

:function_name

The name you gave your application function within your program.

Example

ICAPI application processing

```
PICSDBLookup /index.html d:\ics\api\bin\icsext05.dll:get_pics
```

Initial configuration file setting: None.

DataFilter - Customize the Data Filter step

Use this directive to specify a customized application function you want the server to call during the Data Filter step. This code would provide three application functions:

- An *open* function to perform any initialization prior to processing the data
- A *write* function to process the data
- A *close* function to perform any clean up activities

You can only have one DataFilter active for each instance of the server.

The format of the directive is:

```
DataFilter d:\path\file:function_name:function_name:function_name
```

d:\path\file

The fully qualified file name of your compiled program, including the extension.

:function_name

The name you gave your application function within your program. You will need to supply the name of the open, write, and close functions.

Example

```
DataFilter d:\ics\bin\icsext05.dll:open_data:write_data:close_data
```

Initial configuration file setting: None.

Log - Customize the Log step

Use this directive to specify a customized application function you want the server to call during the Log step. This code would supply logging and other processing you want performed after the connection has been closed.

The format of the directive is:

```
Log request-template d:\path\file:function_name
```

request-template

A template for requests that further determine if your application function is called. The specification can include the protocol, domain and host, can be preceded by a slash (/), and can use an asterisk (*) as a wildcard. For example, /front_page.html, http://www.ics.raleigh.ibm.com, /pub*, /*, and * are all valid.

ICAPI application processing

d:\path\file

The fully qualified file name of your compiled program, including the extension.

:function_name

The name you gave your application function within your program. You must supply the names of the open, write, and close functions.

Example

```
Log      /index.html d:\api\bin\icsextpgm.dll:trans_url
```

Initial configuration file setting: None.

Error - Customize the Error step

Use this directive to specify a customized application function you want the server to call during the Error step. This code would execute only when an error is encountered to provide customized error routines.

The format of the directive is:

```
Error request-template d:\path\file:function_name
```

request-template

A template for requests that further determine if your application function is called. The specification can include the protocol, domain and host, can be preceded by a slash (/), and can use an asterisk (*) as a wildcard. For example, /front_page.html, http://www.ics.raleigh.ibm.com, /pub*, /*, and * are all valid.

d:\path\file

The fully qualified file name of your compiled program, including the extension.

:function_name

The name you gave your application function within your program.

Example

```
Error /index.html d:\ics\api\bin\icsext05.dll:error_rtns
```

Initial configuration file setting: None.

PostExit - Customize the PostExit step

Use this directive to specify a customized application function you want the server to call during the PostExit step. This code will be executed regardless of the return codes from previous steps or other PostExit handlers. It allows you to clean up any resources allocated to process the request.

The format of the directive is:

Java support

```
PostExit    d:\path\file:function_name
```

d:\path\file

The fully qualified file name of your compiled program, including the extension.

:function_name

The name you gave your application function within your program.

Examples:

```
PostExit    \ics\api\bin\icsext05.dll:post_exit
```

Initial configuration file setting: None.

ServerTerm - Customize the Server Termination step

Use this directive to specify a customized application function you want the server to call during the Server Termination step. This code would execute when an orderly shutdown occurs and whenever the server is restarted. It allows you to release resources allocated by a PreExit application function.

The format of the directive is:

```
ServerTerm  d:\path\file:function_name
```

d:\path\file

The fully qualified file name of your compiled program, including the extension.

:function_name

The name you gave your application function within your program.

Example

```
ServerTerm  d:\ics\api\bin\icsext05.dll:shut_down
```

Initial configuration file setting: None.

Servlet API Support - Configure the server for Java servlet API support

Use this directive to configure your server to support Sun's Java servlet API. This will allow you to execute servlets that extend the Java servlet interface and its classes, as defined in the `java.servlet` and `java.servlet.http` packages in Sun's Java Servlet API White Paper (<http://java.sun.com/products/jeeves/CurrentRelease/doc/api.html>). The

Java support

Servlet API classes you need to compile a Java servlet are included in the `icsclass.zip` file in the server's base CGI-BIN directory. If you enable Java servlet support, this file is added to your CLASSPATH.

A servlet is like a server-side applet that runs on a Java thread but does not have a user interface. Servlets are persistent. Although a servlet's `service()` method is executed for each request, servlets are instantiated and initialized just once when they are loaded. This allows servlets to create dynamic output similar to CGI programs.

Using the Java configuration directives, you have the ability to:

- Turn the servlet support on or off
- Specify the number of Java threads to use to process servlet requests
- Name the directory where you keep the servlets
- Choose whether to log servlet messages and the location of the log
- Specify a servlet's initialization parameters

On the Internet Connection Secure Server, servlets cannot be loaded from the network; they must reside on the local system. Servlets run in a separate process than the one in which the Internet Connection Secure Server is running.

Note to servlet writers:

When writing your servlets, you can write your own code for the following servlet class methods (the `service` method is required and the others are optional):

- `init()`
- `service(request,response)`
- `getServletInfo()`
- `destroy()`

All other Java servlet API methods are implemented by the Internet Connection Secure Server for its processing and your use. Be sure you **do not override any other Java servlet API methods** or you will be overriding the server's code.

EnableJavaServletSupport - Enable the server to support Java servlets

Use this directive to specify whether or not you want your server to support Java servlets. If you specify `no`, the servlet support is turned off and all the other Java servlet directives are ignored. If you specify `yes`, the servlet support is turned on, the Java Virtual Machine (JVM) is started, and all the other Java servlet directives are implemented.

The format of this directive is:

```
EnableJavaServletSupport value
```

Example

Java support

```
EnableJavaServletSupport yes
```

Default

```
EnableJavaServletSupport no
```

MaxActiveJavaThreads - Specify threads for request processing

Use this directive to specify the maximum number of Java threads that the server will use to process Java servlets.

The format of this directive is:

```
MaxActiveJavaThreads number
```

Example

```
MaxActiveJavaThreads 25
```

Default:

```
MaxActiveJavaThreads 10
```

ServletLog - Specify a log file for Java servlet messages

Use this directive to specify the name and location of the log file for Java servlet messages. All messages generated by the servlet log method will be written to this file. If you omit this directive, none of the messages generated by the servlet log method will be captured.

Note: This log is not automatically deleted on schedule like other server logs. You will have to monitor and control the size of this log yourself.

The format of this directive is:

```
ServletLog drive:\path\filename
```

Example

```
ServletLog D:\WWW\ics\logs\js-log-12
```

Default

```
ServletLog c:\path\logs\servlet-log
```

c:\path is replaced by the drive and directory where you installed the server. The default directory is C:\WWW.

ServletDir - Specify the directory for Java servlets

Use this directive to specify the name and location of the directory where you will keep all Java servlets.

The format of this directive is:

```
ServletDir  drive:\path\directory
```

Example

```
ServletDir  D:\WWW\ics\servlets\v42\public
```

Default

```
ServletDir  c:\path\public
```

c:\path is replaced by the drive and directory where you installed the server. The default directory is C:\WWW.

Servlet - Specify a servlet's initialization parameters

Use this directive to specify the name and value of the parameters passed to a Java servlet when it is initialized. These must be listed one per line.

The format of this directive is:

```
Servlet servlet-name {
    parm0=value0
    parm1=value1
    parm2=value2
    ...
    parmn=valuen
}
```

Example

```
Servlet MyServlet {
    count=1
    path=d:\w3\list
}
```

Default: None

Proxy server settings

Proxy server settings - Configure server as a proxy

Use the directives described in this section to configure your server as a caching proxy. See “Running your server as a proxy” on page 11 for more information on proxy servers.

CacheDefaultExpiry - Specify default expiration time for files that do not have an expiration date

Use this directive to set a default expiration time for files that the server did not give either an Expires or a Last-Modified header to. You specify a URL template and the expiration time for files with URLs that match the template. You can have multiple occurrences of this directive in the configuration file. Include a separate directive for each template. The URL template must include the protocol. Specify the time value in any combination of months, weeks, days, and hours.

Examples:

```
CacheDefaultExpiry ftp:* 1 month
CacheDefaultExpiry gopher:* 10 days
```

Initial configuration file setting

```
CacheDefaultExpiry ftp:* 1 day
CacheDefaultExpiry gopher:* 1 day
CacheDefaultExpiry http:* 0 days
```

Notice in the above defaults that the default expiration for HTTP is 0. HTTP should be kept at 0 because many script programs don't give an expiration date, yet their output expires immediately. (A value other than 0 may cause problems.)

CacheExpiryCheck - Turn cache expirations off

Use this directive to specify whether you want the server to return cached files that have expired. Specify `Off` for the value if you want the server to be able to return expired files. Use the default value of `On` if you do not want the server to return expired files. Generally, you will not want the server to return expired files. An exception might be if you were demonstrating the server and do not particularly care about the content being returned.

Example

```
CacheExpiryCheck Off
```

Initial configuration file setting

```
CacheExpiryCheck On
```

CacheLastModifiedFactor - Specify fraction of Last-Modified time to be used for determining expiration date

HTTP servers usually give the Last-Modified time for a file, but not the Expires date. Use this directive to have your server approximate the expiration date of these files based on the Last-Modified time. The server uses the Last-Modified date to determine how long it has been since the file was modified. The server multiplies that length of time by the value on the CacheLastModifiedFactor directive. The server uses the result of this calculation to assign the file an expiration date when it caches the file.

Examples:

```
CacheLastModifiedFactor 0.2
```

The above example would cause files modified five months ago to expire after one month.

```
CacheLastModifiedFactor Off
```

The above example would turn this function off.

Initial configuration file setting

```
CacheLastModifiedFactor 1.5
```

The default of 1.5 causes files modified in the past week to be updated in one day.

CacheLimit_1 - Specify lower limit for cached file size

The server uses an algorithm for garbage collection to determine which files to delete. The size of a file is part of the calculation. The size of small files is not taken into account in the calculation. The size of large files is taken into account; the larger the file, the more likely it is to be deleted.

Use this directive to tell the server what should be considered a small file. The value can be specified in bytes (B), kilobytes (K), megabytes (M), or gigabytes (G). It does not matter if you have a space between the number and the value (B, K, M, G).

Example

```
CacheLimit_1 400 K
```

Proxy server settings

Initial configuration file setting

```
CacheLimit_1 20 K
```

CacheLimit_2 - Specify upper limit for cached file size

Use this directive to specify the maximum size of files to be cached. Files larger than this size will not be cached. The value can be specified in bytes (B), kilobytes (K), megabytes (M), or gigabytes (G). It does not matter if you have a space between the number and the value (B, K, M, G).

Example

```
CacheLimit_2 2000 K
```

Initial configuration file setting

```
CacheLimit_2 400 K
```

CacheLockTimeOut - Specify how long a file being cached can remain locked

During retrieval, cache files are locked. If something goes wrong, a locked file may be left hanging. Use this directive to set the amount of time after which the lock on the file can be broken. Specify the time value in minutes (mins).

Note: Set CacheLockTimeOut to a value equal to or greater than OutputTimeOut. The default value of 20 minutes is the same as the default for OutputTimeOut.

Example

```
CacheLockTimeOut 30 mins
```

Initial configuration file setting

```
CacheLockTimeOut 20 minutes
```

CacheNoConnect - Specify stand alone cache mode

Use this directive to specify whether you want the proxy server to retrieve files from remote servers. Use the default value of Off if you want the server to be able to retrieve files from remote servers.

Proxy server settings

Specify `On` if you want the server to run in stand alone cache mode. This means that the server can return only files already stored in its cache. Typically, you would also set the `CacheExpiryCheck` directive to `Off` when running the server in this mode.

Running the server in stand alone cache mode can be useful if you are using the server for demonstrations. If you know all the files you want to use for a demonstration are stored in the cache, then you do not need a network connection.

Example

```
CacheNoConnect On
```

In the above example, the server returns only files stored in its cache.

Initial configuration file setting

```
CacheNoConnect Off
```

CacheOnly - Cache only files with URLs that match a template

Use this directive to specify that only files with URLs that match the given template should be cached. You can have multiple occurrences of this directive in the configuration file. Include a separate directive for each template. The URL template must include the protocol.

Example

```
CacheOnly http://realstuff/*
```

Initial configuration file setting: None

CacheRoot - Specify cache root directory

Use this directive to specify the top directory in the cache hierarchy. The server will create subdirectories within this directory for each cached protocol. It will also create subdirectories under each protocol subdirectory for each remote server.

Example

```
CacheRoot /webcache
```

Initial configuration file setting: None

Proxy server settings

CacheSize - Specify cache size

Use this directive to set the maximum amount of disk space you want the proxy cache to use. If you have plenty of disk space, you may want to substantially increase the 5 M default size. The size of the cache will usually stay below the maximum, but may occasionally grow slightly larger. When the maximum size is reached, the garbage collection process begins. The value can be specified in bytes (B), kilobytes (K), megabytes (M), or gigabytes (G). It does not matter if you have a space between the number and the value (B, K, M, G).

Example

```
CacheSize 50 M
```

Initial configuration file setting

```
CacheSize 5 M
```

CacheUnused - Specify how long to keep unused cached files that match a template

Use this directive to set the maximum amount of time for the server to keep unused cached files with URLs matching a given template. The server deletes unused files with URLs matching the template after they have been cached for the specified time, regardless of their expiration date. You can have multiple occurrences of this directive in the configuration file. Include a separate directive for each template. The URL template must include the protocol. Specify the time value in any combination of months, weeks, days, and hours.

Examples:

```
CacheUnused ftp:* 3 weeks  
CacheUnused gopher:* 3 days 12 hours  
CacheUnused * 4 weeks
```

Initial configuration file setting: None

Caching - Turn proxy caching on/off

Use this directive to enable the caching of files. With caching turned on, the proxy server can store the files it retrieves from other servers in a local cache. The server can then respond to subsequent requests for the same files without having to retrieve them from another servers. This can improve response time.

Example

```
Caching On
```

Proxy server settings

Initial configuration file setting

Caching Off

ftp_proxy - Specify a proxy server for this proxy to connect to for FTP requests

If your proxy server is part of a chain of proxies, use this directive to specify the name of another proxy that this server should contact for FTP requests.

Example

```
ftp_proxy http://outer.proxy.server/
```

Initial configuration file setting: None.

Gc - Turn garbage collection on or off

If you have enabled caching, the server uses the garbage collection process to delete files that should no longer be cached. Files are deleted based on their expiration date and other proxy directive values. Use this directive to turn garbage collection on or off. Generally, you would not turn off garbage collection if you have enabled caching. If you do, your cache file could grow beyond the maximum size you set for it.

Assuming garbage collection is turned on, the garbage collection process runs when the cache reaches its maximum size (as specified on the CacheSize directive). The garbage collection process will also run at the time of day specified on the GcDailyGc directive.

Example

```
Gc On
```

Initial configuration file setting

```
Gc Off
```

GcDailyGc - Specify a daily time for garbage collection

Use this directive to specify a particular time of day to run the garbage collection process. Garbage collection occurs automatically when the cache size limit is reached. By specifying a daily time for garbage collection you can also remove cached files before the cache reaches its maximum. Specify the time value in 24:00 hour format. Generally, you would want the garbage collection process to run when your server is not being used much for other things. This is why the default is 03:00.

Proxy server settings

Example

```
GcDailyGc 22:00
```

The above example would start the garbage collection process at 10 PM.

```
GcDailyGc Off
```

The above example would disable daily garbage collection.

Initial configuration file setting

```
GcDailyGc 03:00
```

GcMemUsage - Specify how much memory to use for garbage collection

Garbage collection works best if it can read all cache information into memory at one time. It may not be able to read in the entire cache if your system does not have enough main memory.

Use this directive to specify how much memory garbage collection can use. The value you specify should be approximately the amount of virtual memory that the server may use while performing garbage collection. The amount of memory needed will vary based on dynamic changes such as the directory structure of cached files. Specify the value as a number that represents kilobytes, but do not put a K next to the number.

If garbage collection fails because there is not enough memory on your system, set this directive to a smaller value. If you have plenty of memory to spare, you may want to set this value above the default of 500.

Example

```
GcMemUsage 100
```

The example above might be used for a machine with a small amount of memory.

Initial configuration file setting

```
GcMemUsage 1000
```

gopher_proxy - Specify a proxy server for this proxy to connect to for Gopher requests

If your proxy server is part of a chain of proxies, use this directive to specify the name of another proxy that this server should contact for Gopher requests.

Proxy server settings

Example

```
gopher_proxy gopher://outer.proxy.server/
```

Initial configuration file setting: None.

http_proxy - Specify a proxy server for this proxy to connect to for HTTP requests

If your proxy server is part of a chain of proxies, use this directive to specify the name of another proxy that this server should contact for HTTP requests.

Example

```
http_proxy http://outer.proxy.server/
```

Initial configuration file setting: None.

MaxContentLengthBuffer - Set the size of the buffer for dynamic data generated by the server

Use this directive to set the size of the buffer for dynamic data generated by the server. Dynamic data is output from CGI programs, server-side includes, and API programs. It is data that does not come from a proxy request.

The value can be specified in bytes (B), kilobytes (K), megabytes (M), or gigabytes (G). It does not matter if you have a space between the number and the value (B, K, M, G).

Example

```
MaxContentLengthBuffer 2 M
```

Initial configuration file setting

```
MaxContentLengthBuffer 50 K
```

no_proxy - Connect directly to domains matching templates

Use this directive to specify the domains that you want the server to directly connect to rather than going through a proxy.

Specify the value as a string of domain names or domain name templates. Separate each entry in the string with a comma. Do **not** put any spaces in the string.

Proxy server settings

You specify templates on this directive a bit differently than the way you specify templates on other directives. Most importantly, you **cannot** use the wildcard character (*). What you **can** do is specify a template by including only the last part of a domain name. The server connects directly to any domains that end with a string matching the templates you specify. The following example shows how this works.

Example

```
no_proxy www.someco.com,.raleigh.ibm.com,.some.host.org:8080
```

In the above example the server would not go through a proxy for the following requests:

- Any requests to domains ending with `www.someco.com`,
- Any requests to domains ending with `.raleigh.ibm.com`, such as `blugrass.raleigh.ibm.com` or `keystone.raleigh.ibm.com`
- Any requests to port 8080 of domains ending with `.some.host.org`, such as `myname.some.host.org:8080`. (This would not include requests to any other ports of the same domain, such as `myname.some.host.org`, which assumes the default port 80.)

Initial configuration file setting None.

NoCaching - Do not cache files with URLs that match a template

Use this directive to specify that the server should not cache files with URLs matching the given template. You can have multiple occurrences of this directive in the configuration file. Include a separate directive for each template. The URL template must include the protocol.

Example

```
NoCaching http://joke/*
```

Initial configuration file setting: None.

ProxyAccessLog - Name the path for the proxy access log file

Use this directive to specify the path and file name where you want the server to log access statistics that pertain to proxy requests. By default, the server writes an entry to this log each time it acts as a proxy for a client request. You can use the `NoLog` directive if you do not want to log requests from certain clients. For a description of the `NoLog` directive, refer to “`NoLog` - Suppress log entries for specific hosts or domains matching a template” on page 71.

Proxy server settings

The server starts a new log file each day at midnight if it is running. Otherwise, the server starts a new log file the first time you start it on a given day. When creating the file, the server uses the file name you specify and appends a date extension. The date extension is in the format *Mmmddyyyy*, where *Mmm* is the first three letters of the month; *dd* is the day of the month; and *yyyy* is the year.

It is a good idea to remove old log files, because they can take up a significant amount of space on your hard drive.

Example

```
ProxyAccessLog c:\server\logs\proxylog
```

Initial configuration file setting

```
ProxyAccessLog c:\path\proxy-log
```

c:\path is the value you entered for **Logs directory** at installation. The installation default is *c:\WWW\LOGS*

SocksServer - Specify a Socks server through which the proxy will pass requests

Use this directive to specify the IP address or host name of the Socks server through which this proxy will be passing requests.

Format

```
SocksServer SocksServer SocksNameServer
```

where *SocksServer* is the IP address or the host name of the Socks server to which the proxy should be chained, and *SocksNameServer* is the IP address or host name of the Socks Name Server.

Example

```
SocksServer socks.bcd.com socksname.bcd.com
```

Initial configuration file setting: None.

wais_proxy - Specify a proxy server for this proxy to connect to for WAIS requests

If your proxy server is part of a chain of proxies, use this directive to specify the name of another proxy that this server should contact for WAIS requests.

Example

```
wais_proxy wais://outer.proxy.server/
```

Performance settings

Initial configuration file setting: None.

Performance settings - Define performance settings

Use the directives described in this section to control the performance of your server.

Each time your server receives a request from a client, it uses one or two threads to perform the requested action. (One thread if the server is not performing DNS lookup. Two threads if the server is performing DNS lookup.) If no threads are available, the server holds requests until more threads are available. The `MaxActiveThreads` directive specifies the maximum number of active threads.

If your server is running at maximum capacity on a sustained, non-stop basis, the amount of virtual memory used increases. This increase is temporary and is alleviated as the number of requests decline and the server catches up on servicing requests.

You can lower the amount of virtual memory used by lowering the `MaxActiveThreads` setting. A good starting point would be half of your current `MaxActiveThreads` setting. Keep in mind when you're lowering `MaxActiveThreads` that, when no threads are available, the server holds requests until more threads are available.

Use the `ServerPriority` directive to specify the priority your operating system gives to the server.

If you see a problem with your performance with your server being too slow, it could be related to any of the following:

- Your network speed
- The traffic on your LAN
- The number of clients requesting from your server
- The number of threads set on your server

Use the `CacheLocalFile` directive to load your most popular files into the servers memory at startup time. You can specify the maximum amount of memory and the maximum number of files for caching with the `CacheLocalMaxBytes` and `CacheLocalMaxFiles` directives.

Use the `PersistTimeout` and the `MaxPersistRequest` directives to specify the characteristics of a persistent connection. A persistent connection allows the server to accept multiple requests and to send responses over the same TCP/IP connection. Overall throughput is increased because the server does not have to establish a separate TCP/IP connection for each request and response. Also, the TCP/IP

Performance settings

connection is used more efficiently because a client can make multiple requests without waiting for the response to each request.

CacheLocalFile - Specify files you want to load in memory at start up

Use this directive to specify the names of files you want to load into the server's memory each time you start the server. You can have multiple occurrences of this directive in the configuration file. Include a separate directive for each file you want to load into memory.

By keeping your most frequently requested files loaded in the server's memory, you can improve your server's response time for those files. For example, if you load your server's welcome page into memory at startup, the server can handle requests for the page much more quickly than if it had to read the file from a disk. Keep in mind that for each file you load into memory, you are making that amount of memory unavailable for other uses.

Before responding to a request for a file that is stored in memory, the server checks to see if the file has changed since the server was started. If the file has changed, the server responds to the request with the updated file and deletes the old version from its memory. To load the new file into memory, you need to restart the server.

Notes:

1. You can use an asterisk (*) as a wildcard character on the file names.
2. File name matching is not recursive. Only files in the specified directory will be cached. No files in subdirectories are affected.

Example: To cache a specific file

```
CacheLocalFile d:\www\html\index.html
```

To cache all .html files in the powerco directory

```
CacheLocalFile d:\www\powerco\*.html
```

Default: The default configuration includes CacheLocalFile directives for the HTML and graphics files that make up the server's Front Page.

CacheLocalMaxBytes - Specify maximum amount of memory to use for file caching

Use this directive to specify the maximum amount of memory you want to allow for file caching. You can specify the memory in kilobytes (K) or megabytes (M). You must still specify the files that you want cached with the CacheLocalFiles directive.

Performance settings

Note: CacheLocalMaxBytes can help limit your cache size when you are using the wildcard character to specify the files on the CacheLocalBytes directive.

Example

```
CacheLocalMaxBytes 5K
```

Default

```
CacheLocalMaxBytes 2M
```

CacheLocalMaxFiles - Specify the maximum number of files for caching

Use this directive to specify the maximum number of files you want to be cached at one time. You must still use the CacheLocalFiles directive to indicate which files you want cached.

Note: CacheLocalMaxFiles can help limit your cache size when you are using the wildcard character to specify the files on the CacheLocalFiles directive.

Example

```
CacheLocalMaxFiles 150
```

Default

```
CacheLocalMaxFiles 200
```

LiveLocalCache - Specify whether the cache is updated when a cached file is modified

Use this directive to specify whether or not the cache is updated when a cached file is modified. Specify ON if you want users requesting a cached file to get the file with the latest updates. OFF is the high performance setting.

Initial configuration file setting

```
LiveLocalCache off
```

Program default setting

```
LiveLocalCache off
```

MaxActiveThreads - Specify the maximum number of threads to have active

Use this directive to set the maximum number of threads that you want to have active at one time. If the maximum is reached, the server holds new requests until another request finishes and threads become available. Generally, the more power your machine has, the higher the value you should use for this directive. If your machine starts to spend too much time on overhead tasks, such as swapping memory, try reducing this value. The recommended value is 40. Unpredictable results may occur if this value is exceeded.

Example

```
MaxActiveThreads 49  
MaxActiveThreads 35
```

Initial configuration file setting

```
MaxActiveThreads 40
```

ServerPriority - Specify the priority you want your server to have on your system

Use this directive to specify a priority class for the Internet Connection Secure Server. The operating system uses priority classes to determine which processes have priority over others.

Valid values are:

- 0 - no priority
- 1 - maximum priority as a normal process
- 2 - maximum priority as a foreground server process

You may want to use a value of 0 if the machine your server is running on is also processing other types of requests.

Example

```
ServerPriority 0
```

Default:

```
ServerPriority 1
```

Performance settings

MaxPersistRequest - Specify the maximum number of requests to receive on a persistent connection

Use this directive to specify the maximum number of requests the server should receive on a persistent connection. When determining this number, be sure to consider the number of images used in your pages. Each image requires a separate request.

The format of this directive is:

```
MaxPersistRequest number
```

number is the number of requests the server should receive for a persistent connection.

Initial configuration file setting:

```
MaxPersistRequest 5
```

Program default setting:

```
MaxPersistRequest 5
```

PersistTimeout - Specify time to wait for the client to send another request

Use this directive to specify the amount of time the server should wait between client requests before cancelling a persistent connection.

The server uses a different timeout, the input timeout, to determine how long to wait for the client to send the first request after the connection is established. For more information on the input timeout, see “InputTimeout - Specify input timeout setting” on page 116.

After the server sends its first response, it uses the persistent timeout to determine how long it should wait for each subsequent request before cancelling the persistent connection.

The format of this directive is:

```
PersistTimeout time
```

time can be any valid time increment, but usually will be seconds or minutes

Initial configuration file setting:

```
PersistTimeout 1 minutes
```

Performance settings

Program default setting:

PersistTimeout 1 minutes

UseACLs - Specify whether ACL files will be checked

Use this directive to specify whether the ACL files will be checked for file protection. Set this directive to never or protect only for better server performance. The format of the directive is:

UseACLs *setting*

The *setting* can have a value of always, protect only, or never.

always

The server will always look for an ACL file on every file request.

protect only

The server will only look for an ACL file when the file request is for a file that is covered by a protection statement.

never

The server will never look for an ACL file on a file request.

Example

```
UseACLs protectonly
```

Initial configuration file setting

```
UseACLs always
```

UseMetaFiles - Specify whether meta files will be used

Use this directive to specify whether the meta files used by the server. Set this directive to off for better server performance. The format of the directive is:

UseMetaFiles *setting*

The *setting* can have a value of on or off.

on The server will always use meta files.

off The server will not use meta files.

Example

```
UseMetaFiles off
```

System Management

Initial configuration file setting

UseMetaFiles on

System Management - Define system management settings

Use the directives described in this section to define settings used to monitor and manage the health, throughput, and activity of your servers.

SNMP - Enabling and disabling SNMP support

Use this directive to enable or disable SNMP support. The form of the directive is:

SNMP *setting*

The *setting* can have a value of on or off.

on SNMP support is turned on.

off SNMP support is turned off.

Example

SNMP on

Initial configuration file setting

SNMP off

Program default setting

SNMP off

SNMPCommunityName - Providing a security password for SNMP

Use this directive to define the password between the Internet Connection Secure Server DPI subagent and the SNMP agent. The SNMP community name authorizes a user to view the performance variables monitored by SNMP for a particular community of servers. The system administrator defines which variables from which servers can be viewed when a password is entered. If you change this value, you must also add the new community name to the SystemView Agent SNMP Configuration Notebook.

For more information about creating a community name, refer to “Providing a security password for SNMP” on page 233.

The form of the directive is:

```
SNMPCommunityName com_nam
```

Example

```
SNMPCommunityName public
```

Initial configuration file setting

```
SNMPCommunityName public
```

Program default setting

```
SNMPCommunityName public
```

WebMasterEmail - Creating an e-mail address to receive SNMP problem reports

Use this directive to create an e-mail address to receive SNMP problem reports. The default mail address is webmaster.

The form of the directive is:

```
WebMasterEmail webmastermailaddress
```

Example

```
WebMasterEmail webmaster@computer.com
```

Initial configuration file setting

```
WebMasterEmail webmaster
```

Chapter 4. Customizing logs and reports

Your server can create several types of logs:

- Access logs
- Agent logs
- Error logs
- Cache access logs, if your server is a caching proxy
- CGI error logs
- Referer logs

For access logs, you can set filters and create reports that help you analyze the information in the access logs. You can also set maintenance options. For error logs, you can set certain maintenance options. For the other types of logs, you can specify the name of the log and where you want it to be filed.

This chapter explains how to tailor the access and error logs to meet your needs, and also how create customized reports from the information in the logs.

| | |
|---|-----|
| Tailoring the logs your server keeps | 153 |
| Specifying global settings for all logs | 155 |
| Specifying options for the access, agent, and referer logs | 155 |
| Specifying options for the error logs | 158 |
| Sample scenario for configuring log files | 160 |
| Specifying the path for the proxy server's cache access log | 161 |
| Tailoring the reports your server creates | 161 |
| Overview of report templates | 162 |
| Creating a report template | 164 |
| Viewing reports | 165 |
| Sample scenarios for configuring reports | 165 |

Tailoring the logs your server keeps

The server creates many types of logs. Each day at midnight, the server closes the logs for that day and creates new logs.

The server logs activity in the **access log** files and stores them on the hard drive each night. At midnight each night, the server closes the current access log and creates a new access log file for the coming day. The access log contains entries for page request made to the server.

Customizing Logs and Reports

For each access request your server receives, an entry is made in the access log showing:

- What was requested
- When it was requested
- Who requested it
- The method of the request
- The type of file that your server sent in response to the request
- The return code, which indicates whether the request was honored

The server can also create an **agent log** and a **referer log**. The agent log indicates which Web browser was used to access a Web page. The referer log identifies the Web page that referred (or linked to) the requested Web page. By default the server writes an entry to the agent and referer logs each time a client sends the server a request. For every entry made in the access log:

- The agent log has a corresponding entry that indicates the browser used to display the page or file requested by the client
- The referer log has a corresponding entry that indicates the referring page

The server creates an **error log** that includes errors encountered by your server's clients, such as timing out or not getting access.

The server also creates a **CGI error log** that logs standard error output (stderr) from CGI programs.

If your server is running as a proxy, the server can create two different types of logs:

- A **proxy access log**, which contains access requests for files that come from the proxy server
- A **cache access log**, which contains access requests for files that come from the proxy server's cache

This section describes how to set up the logs to suit your particular needs. If you are satisfied with the default setting for an option, you can skip the step. Look at the sections that apply to you:

1. If you want to change the default global settings, read "Specifying global settings for all logs" on page 155.
2. If you want to set up your access, agent, and referer logs, read "Specifying options for the access, agent, and referer logs" on page 155.
3. If you want to set the path and the maintenance options for your error and CGI error logs, read "Specifying options for the error logs" on page 158.
4. If you want to set the path for your proxy server's cache access log, read "Specifying the path for the proxy server's cache access log" on page 161.

Note: You can change the default settings for the logs either by using the online Configuration and Administration forms or by manually editing the directives in the configuration file.

Specifying global settings for all logs

In most cases, you will want to accept the default global settings, which apply to all logs.

If you plan to use the reporting functions described under “Tailoring the reports your server creates” on page 161, you must accept the default file format, common.

If you want to have log information sent to the Internet Connection Secure Server window in addition to sending it to the log files, you must change the default.

To change the global settings, we recommend that you specify them on the **Global Log File Configuration** form.

Defaults: Common file format, which is used by most Web servers and local time format are used. By default, access log information is written only to the access log, (not the syslog) and error log information is written only to the error log, (not the syslog).

Directives

- For time stamp, edit the LogTime directive.
- For file format, edit the LogFormat directive.
- For logging to the Internet Connection Secure Server window, edit the LogToGUI directive.

You cannot change whether the logs are written nightly.

Specifying options for the access, agent, and referer logs

This section describes the following tasks:

- “Specifying the access, agent, and referer log paths”
- “Choosing log maintenance options for the access, agent, and referer logs” on page 156
- “Setting filters for the access, agent, and referer logs” on page 157

Specifying the access, agent, and referer log paths

From the **Access Log File Configuration** form, you can specify the path and name of the directory where you want to place the access, agent, and referer log files. The directory must be on an HPFS-formatted drive.

Customizing Logs and Reports

Defaults: We strongly recommend that you accept the default path, which is the value you entered for **Logs directory** at installation.

Directives

1. For the access log path, edit the AccessLog directive.
2. For the agent log path, edit the AgentLog directive.
3. For the referer log path, edit the RefererLog directive.

Choosing log maintenance options for the access, agent, and referer logs

With the log maintenance options, you can specify how to handle the accumulation of daily logs for days past.

You can choose whether you want to keep old logs, remove logs after they reach a certain age and/or a collective size, or run your own program at midnight each night to handle old logs. Note that the “collective size” is the collective size of all access logs only (not combined with agent and referer logs), or of all agent logs only (not combined with access and referer logs), or of all referer logs only (not combined with access and agent logs).

To reduce the space the access, agent, and referer logs require, you can specify that the logs be automatically removed, based on the age of the log and/or the collective size of the logs.

If you are interested in running your own backup program to store the logs, you can specify a user exit. In this case, you specify the path to your program and the parameters to pass to your program. The server appends to this information the path to the logs on your hard drive.

We recommend you define these options on the **Access Log File Configuration** form, but you can edit the configuration file to include the appropriate directives. The settings you specify on the **Access Log File Configuration** form apply to agent and referer logs, as well.

Defaults: By default, all access, agent, and referer log files are kept on the hard drive at the path location you specify on the **Access Log File Configuration** form. (or the AccessLog, AgentLog, and RefererLog directives).

Directives: The directives you specify for access logs apply to agent and referer logs, as well.

- To keep access log files, no directive is required. AccessLogArchive none is the default.
- To remove access log files based on age, edit these directives:
 - AccessLogArchive purge

Customizing Logs and Reports

- AccessLogExpire *number-of-days*
- To remove access log files based on collective size, edit these directives:
 - AccessLogArchive purge
 - AccessSizeLimit *number-of-megabytes*
- To run a user exit, edit the AccessLogArchive userexit directive.

For details on these directives, refer to “AccessLogArchive - Remove existing access, agent, or referer log files or run a user exit” on page 55.

Setting filters for the access, agent, and referer logs

For the access log, you can set filters so that the access, agent, and referer logs includes only the information you are interested in.

To improve your ability to use the information included in the access, agent, and referer log files, you can filter out extraneous information so that the log includes only information that is meaningful to you. You filter out information by excluding entries that match a particular pattern. We recommend you define these options on the **Access Log File Configuration** form, but you can edit the configuration file to include the appropriate directives for the filters you want to set. You can specify filters based on any of the following:

- URL (directories and/or files)
- IP address or host name
- Method
- MIME type
- Return code

Note: Keep in mind that information filtered out from the access log will not show up in any access report and will not be available for future use.

Here are some reasons for controlling what gets logged.

To reduce the size of the logs: You might be interested in reducing the number of entries in an access log to include only meaningful access requests. Access log files can grow rapidly, since by default they contain entries for all access requests for GIF images, HTML pages, and so on. You might want to configure your access logs so that they include log entries for access requests to HTML pages, but not for the access requests for the GIF images that the HTML contains. For example, an HTML page might include several GIF images, which can cause the size of the access log to grow rapidly.

To collect information about external hits only: You might be interested only in who is accessing your server from outside your company. In this case, you would filter out access requests that originate from internal company IP addresses.

Customizing Logs and Reports

To gather information about who is accessing a particular Web site: To help you determine the size of the audience for a particular Web site, you might want to create an access log that shows only the hits to one URL.

Default: By default, everything is logged to the access log, unless you choose to filter out (exclude) something. From the **Access Log File Configuration** form, you can specify what you want to filter out from the access log. You do not need to fill in the entire form.

Scroll to the **Exclusions from the Access log** section of the form. Choose which of the following you want to base filtering on:

- Directories and files
- Host names or IP addresses
- Methods (GET, PUT, POST, DELETE)
- MIME types (images, text, applications, audio, video, multimedia, and other)
- Return code (success, redirection, client error, and server error)

If you want to filter based on directories and files or IP addresses and host names, you need to update the index list on the **Access Log File Configuration** form. You can insert or remove entries in the list to specify what you want filtered out. To exclude entries based on methods, MIME types, or return codes, click the boxes that describe what you want to filter out.

When you have finished specifying what you want to exclude on the **Access Log File Configuration** form, click **Apply** to have the filters take effect.

Directives

- To filter out files or directories that match a particular pattern, edit the `AccessLogExcludeURL` directive.
- To filter out entries of a particular method, edit the `AccessLogExcludeMethod` directive.
- To filter out entries of a particular MIME type, edit the `AccessLogExcludeMimeType` directive.
- To filter out entries receiving a particular set of return codes, edit the `AccessLogExcludeReturnCode` directive.

Specifying options for the error logs

This section describes the following tasks:

- “Specifying the path for the error and CGI error logs” on page 159
- “Choosing log maintenance options for the error and CGI error logs” on page 159

Specifying the path for the error and CGI error logs

From the **Error Log File Configuration** form, you can specify the path and name of the directory where you want to place the error and CGI error log files. The directory must be on an HPFS-formatted drive. As an alternative, you can specify this information manually by editing the directive listed below.

Defaults: We strongly recommend that you accept the default path, which is the value you entered for **Logs directory** at installation.

Directives: For path, edit the ErrorLog directive.

Choosing log maintenance options for the error and CGI error logs

You can choose whether you want to keep old logs, remove logs after they reach a certain age and/or a collective size, or run your own program at midnight each night to handle old logs. Note that the “collective size” is the collective size of all error logs only (not combined with CGI error logs) or all CGI error logs only (not combined with error logs).

To reduce the space error and CGI error logs require, you can specify that the logs be automatically removed, based on the age of the log and/or the collective size of the logs.

If you are interested in running your own backup program to store the logs, you can specify a user exit. In this case, you specify the path to your program and the parameters to pass to your program. The server appends to this information the path to the logs on your hard drive.

Default: By default, all error and CGI error log files are kept on the hard drive at the path location you specify on the **Error Log File Configuration** form (or the ErrorLog directive.)

We recommend you define these options on the **Error Log File Configuration** form, but you can edit the configuration file to include the appropriate directives. The settings you specify on the **Error Log File Configuration** form apply to CGI error logs, as well.

Directives: The directives you specify for error logs apply to CGI error logs, as well.

- To keep error log files, no directive is required. ErrorLogArchive none is the default.
- To remove error log files based on age, edit these directives:
 - ErrorLogArchive purge
 - ErrorLogExpire *number-of-days*

Customizing Logs and Reports

- To remove error log files based on collective size, edit these directives:
 - `ErrorLogArchive purge`
 - `ErrorSizeLimit number-of-megabytes`
- To run a user exit, edit the `ErrorLogArchive userexit` directive.

For details on these directives, refer to “ErrorLogArchive - Remove existing error or CGI error log files or run a user exit” on page 67.

Sample scenario for configuring log files

In the following example, you have just purchased and installed the Internet Connection Secure Server. You want to set up your server to log access information and error information in the following ways:

- You want both types of logs to use a local time stamp and a common log file format.
- You want the access logs to be purged when they are more than 30 days old and/or when they reach a collective size of 25 megabytes. You do not want following requests to be logged to the access log:
 - Requests for GIF images
 - Requests from hosts with IP addresses that match 9.67.*.*
 - Redirection requests (requests that yield a return code between 300 and 399)
- You do not want the error logs to be purged.

You can specify these criteria by using the Configuration and Administration forms, or by updating the configuration file directives.

Forms

- Use the **Global Log File Configuration Settings** form to set the time and file format
- Use the **Access Log File Configuration** form to:
 - Set the interval for removing old access logs
 - Exclude the MIME type of images/GIF
 - Exclude requests from hosts with IP addresses in the pattern 9.67.*.*
 - Exclude requests that yield a return code between 300 and 399
 - Specify the path for the access log file
- Use the **Error Log File Configuration** form to indicate that you want to keep the error log files.

Directives: For the above scenario, update the configuration file as follows:

Customizing Logs and Reports

| | |
|----------------------------|-----------|
| LogFormat | Common |
| LogTime | LocalTime |
| AccessLogArchive | purge |
| AccessLogExpire | 30 |
| AccessLogSizeLimit | 25 |
| AccessLogExcludeURL | *.gif |
| NoLog | 9.67.*.* |
| AccessLogExcludeReturnCode | 300 |
| ErrorLogArchive | none |

Specifying the path for the proxy server's cache access log

If the server is running as a proxy, you can log requests to the cache separately from other requests.

From the **Access Log File Configuration** form, you can specify the path and file name where you want the server to put access requests that are satisfied from the proxy server's cache. As an alternative, you can specify this information manually by editing the directives listed below.

For more information, refer to “CacheAccessLog - Specify the path for the cache access log files” on page 65.

Directives: For the cache access log path, edit the CacheAccessLog directive, in addition to those described under “CacheAccessLog - Specify the path for the cache access log files” on page 65.

Tailoring the reports your server creates

Several types of files are used in report creation. These files are located in the reports root directory

- Access log data file, named *access.mmdyyyy*
- Report data file, named *template_name.txt*
- Report template file, named *template_name.log*
- Report template list, named *Templates*

The **access log data file** (*access.mmdyyyy*) corresponds to the *httpd-log.mmdyyyy* file that is in the access log directory. For each entry in the access log file, there is an entry in the access log data file. The format of the data in the access log data file is:

IP_address URL seconds_since_1970 number_of_bytes_transferred method code

Customizing Logs and Reports

The **report data file** (*template_name.txt*) includes data from all the access log data files filtered through the template definition. For example, for the template “Top100,” there is a data file called Top100.txt, which corresponds to all the access.*mmdyyy* files filtered through the Top100 template definition and saved to the Top100.txt data file. The format of the data in the report data file is:

```
IP_address URL seconds_since_1970 number_of_bytes_transferred method code
```

The **report template file** (*template_name.log*) is the definition of the template. For example, the template “Top100” has a Top100.log file associated with it, which consists of the Top100 template definition in an ASCII file.

The **report template list** (the file name is Templates) is a list of all the templates that have been defined.

Before you can see a report, you must create a report template that is stored as the report template file. For instructions on how to create a report template, see “Overview of report templates.”

Your server creates reports that include some or all of the contents of the access logs. At midnight each night, the server closes the current access log and creates a new access log file for the coming day. Reports are generated at that time using the access log that was just closed. Reports can also be generated for logs that have been archived.

Note: Your server might not be able to create a report for a very large access log file (60 MB, for example). If your system encounters a problem while trying to generate a report, the cause might be an access log that is too large. To generate reports against very large access log files, try increasing the RAM and (or) the swapper file space on your system. A short-term solution to this problem is to turn off report generation by renaming the htlogrep executable file.

Note: If you have specified the CacheAccessLog directive or if you have indicated on the **Access Log File Configuration** form a path and file name for the proxy server's cache access log, your reports *will not* contain access requests for cached files. If you do not have a cache access log, access requests for a proxy server are logged in the access log and can be included in an access report.

Overview of report templates

You control what is included in reports by filtering out entries that match a particular pattern. These options are defined either by using the Configuration and Administration forms or by editing the configuration file. You can use the forms or the configuration file to specify filters based on any of the following:

- URL (directories and files)
- IP address or host name
- Method
- Return code

Customizing Logs and Reports

The contents of the report are governed by the following factors:

- The log file filters that were in effect when the log file was created
- The report filters that are in effect when the report is created

At report creation time, you control only the report filters that are currently in effect. You cannot include in the report entries that were filtered out from the log file.

You can specify report filters in two ways; you must decide which is easier in your situation.

- If you want to include in your report only a small percentage of the contents of the access log, it might be simplest to specify what to **include**, rather than everything you want to exclude. Think of specifying one or a few include filters as a shortcut to specifying many exclude filters.
- If you want to include in your report most of the contents of the access log, it might be simplest to specify what to **exclude**, rather than everything you want to include.

In some cases, you will find it simplest to specify both include and exclude filters. In this case, it is important to understand how include and exclude filters work together. The include filters are processed first. The report function searches the access log to find all entries that match any include filter patterns. If several include filters are specified, the filters act as OR Boolean expressions. In other words, entries that match at least one of the include filters are included.

The exclude filters are processed after all include filters have been processed. The exclude filters work only on the set of entries that have been already included by the include filters. For clarification, refer to the examples under “Sample scenarios for configuring reports” on page 165.

The include and exclude filters are specified on the **Access Log Report Template Creation** form or can be specified with the AccessReport directives.

Here are some reasons for controlling what gets reported.

To reduce the scope of the report: You might be interested in reducing the scope of the report so that it includes only a portion of what is contained in the log. You can even create several reports, each to gather different information from the same log. You might want to create your report template so that it includes log entries for access requests to HTML pages, but not for the access requests for the GIF images that the HTML contains.

To collect information about external hits only: You might be interested only in who is accessing your server from outside your company. In this case, you would filter out access requests that originate from internal company IP addresses.

To gather information about who is accessing a particular Web site: To help you determine the size of the audience for a particular Web site, you might want to create a report that shows only the hits to one URL.

Customizing Logs and Reports

To discover the top Web pages on your server: To help you determine the popularity of a particular Web site, you filter out everything in the report, except for the most visited Web pages.

Creating a report template

Before you create a report, you must modify or create a report template that outlines what you want the report to contain. To start configuring a report template, choose one of the following options from the **Access Log Report Templates** form:

- If you want to create a report that is very different from your existing reports, choose **Create a new template**.

If you have never created a report, before it might be easier to copy an existing template and edit the copy, rather than creating a new template. To use an existing report template as the basis of a new report, choose **Copy existing template**.

- To update an existing report template, choose **Edit existing template**.
- To delete an existing report template, choose **Delete existing template**.

When you choose **Create a new template** or **Edit existing template**, the **Access Log Report Template Creation** form appears.

When you have finished filling in the form, choose **Apply**.

On this form, you can specify some or all of the following:

- Basic settings, such as the report name, description, and the number of entries to include in the report (for example, the top 10)
- Entries from the access log that you want to exclude from the report (based on directories and file names, host names or IP addresses, methods, or return codes)
- Entries from the access log that you want to include (as a shortcut to many excludes)

Specifying entries to include is a shortcut to specifying many, many excludes. When you want to include only a few types of entries in the report, it is easier to specify what to include rather than excluding nearly everything. For example, if you want to include only access requests for a particular URL, you would include that URL, rather than excluding all the others.

The **Access Log Report Template Creation** form allows you to specify includes and excludes. It is important to understand how includes and excludes affect each other.

- In the index list, if nothing is listed with a Filter Action of "Include," the entire log is included in the report, minus the entries that are excluded.
- If anything is listed with a Filter Action of "Include," the report will contain only the information that is to be included, minus the entries that are excluded.

Customizing Logs and Reports

If you are using the **Access Log Report Templates** form, you see at the bottom of the form the field **Report root directory**. This field is filled in with a default directory. We recommend that you accept the default, rather than changing it. If you choose to change the default, you will need to create a new directory for the path you specify, give the directory the appropriate permissions and add a PASS statement to enable the server to honor requests to store reports in that directory.

Viewing reports

To see a report, from the Configuration and Administration Form page, choose **Access reports**. From there, select the following options:

- Report template
- One of the following filters:
 - URL (directories and files)
 - IP address or host name
 - Method
 - Return code
- Date range

The report is created and displayed after you select the options.

Sample scenarios for configuring reports

You have just purchased and installed the Internet Connection Secure Server and you want to set up your server to automatically generate four different access log reports.

Sample report: Top 100 page hits

You are interested in knowing which Web pages on your server get the most attention. You decide to create a report that meets the following criteria:

- The name of the report is “Top100”
- The report description says “Top 100 page hits”
- Requests for GIF images are not included

You can specify these criteria by using the Configuration and Administration forms, or by updating specific directives in the configuration file.

Forms

- 1** From the Access Log Report Templates form, choose **Create a new template**. Select before in the **List it** field.

Customizing Logs and Reports

- 2 Change nothing in the **Root report directory** field.
- 3 Choose **Apply**.
- 4 On the Access Log Report Template Creation form, for **Report name**, type Top100.
- 5 For **Report description**, type Top 100 page hits--Report on the top 100 items accessed
- 6 For **Report on top**, type 100
- 7 Scroll down the form.
- 8 Choose **Add and Include Directories/Files listed below**
- 9 In the text box, type *.GIF.
- 10 Scroll down to the end of the form and choose **Apply**

Directives

```
AccessReportTemplate Top100 {
    AccessReportDescription    Top 100 page hits
    AccessReportTopList        100
    AccessReportExcludeURL     *.GIF
}
```

Sample report: PUT requests to beta subdirectory

You are running a site that distributes beta-level software and are interested in knowing what is being written to the beta directory and who is requesting PUT access. You decide to create a report that meets the following criteria:

- The name of the report should be “BetaPuts”
- The report description should say “PUT requests to beta subdirectory”
- The report should include only requests for PUT access to the beta subdirectory, which is located at /www/beta

You can specify these criteria by using the Configuration and Administration forms, or by updating specific directives in the configuration file.

Forms

- 1 From the **Access Log Report Templates** form, choose **Create a new template**. Select before in the **List it** field.
- 2 Change nothing in the **Root report directory** field.
- 3 Choose **Apply**.

Customizing Logs and Reports

- 4 On the **Access Log Report Template Creation** form, for **Report name**, type BetaPuts.
- 5 For **Report description**, type PUT requests to beta subdirectory.
- 6 Scroll down the form.
- 7 Choose **Add** and **Include Directories/Files listed below**
- 8 In the text box, type /www/beta/*.
- 9 Scroll down the form.
- 10 Under **Exclude following Methods...** choose **GET, POST, and DELETE**.
- 11 Scroll down to the end of the form and choose **Apply**.

Directives

```
AccessReportTemplate BetaPuts {
    AccessReportDescription    PUT requests to beta subdirectory
    AccessReportIncludeURL     /www/beta/*
    AccessReportExcludeMethod  GET
    AccessReportExcludeMethod  POST
    AccessReportExcludeMethod  DELETE
}
```

Sample report: Accesses, excluding beta subdirectory and alpha7 requests

You are interested in knowing which files on your server are being accessed. However, you want to exclude beta programs, which have files located in the beta subdirectory. You also do not want to include any information on the “Alpha7” project, which has pages named Alpha7*.* in various subdirectories. You decide to create a report that meets the following criteria:

- The name of the report should be “NoBetaAlpha7”
- The report description should say “Accesses, except beta subdirectory and alpha7”
- The report should include all accesses, except those to the beta subdirectory at \www\beta or those files with the name alpha7 anywhere on the server.

You can specify these criteria by using the Configuration and Administration forms, or by updating specific directives in the configuration file.

Forms

- 1 From the **Access Log Report Templates** form, choose **Create a new template**. Select before in the **List it** field.
- 2 Change nothing in the **Root report directory** field.

Customizing Logs and Reports

- 3 Choose **Apply**.
- 4 On the **Access Log Report Template Creation** form, for **Report name**, type NoBetaAlpha7.
- 5 For **Report description**, type Accesses, except beta subdirectory and alpha7.
- 6 Scroll down the form.
- 7 Choose **Add and Exclude Directories/Files listed below**
- 8 In the text box, type
/www/beta/*
/www/beta/alpha7*.*
- 9 Scroll down the form.
- 10 Scroll down to the end of the form and choose **Apply**.

Directives

```
AccessReportTemplate NoBetaAlpha7 {  
    AccessReportDescription    Accesses, excluding beta and alpha7 requests  
    AccessReportExcludeURL     /www/beta/*  
    AccessReportExcludeURL     alpha7*.*  
}
```

Sample report: Accesses for department server and for the beta subdirectory except Alpha7*.* files

Your server is a department server and you want to know the access requests for that server. You also want to know access requests for the beta subdirectory, but you are not interested in knowing access requests for any Alpha7*.* files. You decide to create a report that meets the following criteria:

- The name of the report should be “DeptServer_Beta-NotAlpha7”
- The report description should say “Accesses for Department Server and accesses for beta subdirectory, excluding Alpha7 files”
- The report should include all accesses to the IP address that represents the department server.
- The report should include all access requests to /www/beta/*.
- The report should exclude all access requests to Alpha7*.* files.

You can specify these criteria by using the Configuration and Administration forms, or by updating specific directives in the configuration file.

Customizing Logs and Reports

Forms

- 1 From the **Access Log Report Templates** form, choose **Create a new template**. Select Add in the **List it** field.
- 2 Change nothing in the **Root report directory** field.
- 3 Choose **Apply**.
- 4 On the **Access Log Report Template Creation** form, for **Report name**, type DeptServer_Beta-NotAlpha7
- 5 For **Report description**, type Accesses for Department Server and accesses for beta subdirectory, except Alpha7 files.
- 6 Scroll down the form.
- 7 Choose **Add** and **Exclude Directories/Files listed below**
- 8 In the text box, type
/www/alpha
/www/gamma
/www/delta
Alpha7*.*
- 9 Choose **Add** and **Include host names listed below**
- 10 In the text box, type 9.67*.*
- 11 Scroll down to the end of the form and choose **Apply**.

Directives

```
AccessReportTemplate {
    AccessReportTemplate      DeptServer_Beta-NotAlpha7
    AccessReportDescription   Accesses for Department Server
                             and accesses for beta subdirectory,
                             excluding Alpha7 files
    AccessReportExcludeURL    /www/alpha
    AccessReportExcludeURL    /www/gamma
    AccessReportExcludeURL    /www/delta
    AccessReportExcludeURL    Alpha7*.*
    AccessReportIncludeHostName 9.67*.*
}
```

Customizing Logs and Reports

Chapter 5. Customizing your Web site

This chapter describes methods for customizing the appearance of your Web site and includes the following topics:

| | |
|---|-----|
| Displaying page count, date, time, and text on a Web page | 171 |
| Configuration instructions | 171 |
| Options | 172 |
| Using server-side includes to insert information into CGI programs and HTML documents | 178 |
| Considerations for using server-side includes | 178 |
| Preparing to use server-side includes | 178 |
| Format for server-side includes | 179 |
| Directives for server-side includes | 179 |

Displaying page count, date, time, and text on a Web page

This section explains how to use the htcount program to display the following information on a Web page:

Page count

The page counter is incremented each time the Web page is accessed, and the current value is displayed on the Web page.

Date and time

The current date and time are displayed on the Web page.

Text User-specified text is displayed on the Web page.

Configuration instructions

To use the htcount program:

- 1 Uncomment lines in the server configuration file for the functions you are using.

- Page count:

```
# Service /cgi-bin/apicounter* cgi-bin_root/cgi-bin/htcount:HTCounter*
```

- Date and time:

```
# Service /cgi-bin/datetime* cgi-bin_root/cgi-bin/htcount:HTCounter*
```

Customizing Your Web Site

- Text:

```
# Service /cgi-bin/text2gif* cgi-bin_root/cgi-bin/htcount:HTCounter*
```

For *cgi-bin_root*, enter the directory where the server's CGI programs are installed. This directory should contain the file HTCount.dll. The default directory is C:\WWW\CGI-BIN.

2 Create the counter file and initialize the counter.

The Web administrator must create the counter file and initialize the counter to some value, for example, 0. This enables the Web administrator to control access to the Web site and to Web site resources used to display the requested information.

The counter file must be located in the *server_root*\Counters directory, and the server needs to have write access to the counter file.

Sample files are located in the *server_root*\HTML directory, or you can go to the following URLs:

- To view the sample counter page, go to URL
<http://your.server.name/counter1.html>.

This page includes an explanation of error messages that may be issued by the htcount program when you begin using the counter file.

- To view the sample color page, go to URL
<http://your.server.name/counter2.html>.

For *your.server.name*, enter the fully qualified name of your host, for example, **<http://www.ibm.com>**.

3 Insert lines in the Web page HTML file for the functions you are using.

- Page count:

```

```

- Date and time:

```

```

- Text:

```

```

For *your.server.name*, enter the fully qualified name of your host, for example, **<http://www.ibm.com>**.

For *counter_name*, enter the name of the counter file you created, for example, **cntfile.cnt**.

Options

This section describes the options you can use for displaying the page count, date, time, and text on your Web page.

Customizing Your Web Site

Notes:

1. Defaults are shown in **bold** letters.
2. Option names and values are **not** case-sensitive.
3. Use an ampersand (&) to separate options.

For example, to display a page counter with a foreground color of blue and a background color of white, use the following URL:

```

```

4. The *RRGGBB* color option allows you to specify the color using a hexadecimal color code, where *RR*, *GG*, and *BB* are the hexadecimal digits that specify the Red, Green, and Blue values of the color. Examples of color values are:

| | |
|--------|--------|
| Black | 000000 |
| Red | FF0000 |
| Orange | FFA500 |
| Green | 00FF00 |
| Blue | 0000FF |
| Yellow | FFFF00 |
| White | FFFFFF |

For example, to display a page counter with a foreground color of yellow, you would use the following URL:

```

```

For colors demonstrated online, go to URL <http://your.server.name/counter2.html> or Lem Apperson's Color Index at URL <http://www.infi.net/wwwimages/colorindex.html>.

Common options

FG=*color*

where *color* specifies the foreground color, and can be:

Black
White
Red
Green
Blue
RRGGBB

BG=*color*

where *color* specifies the background color, and can be:

Transparent
Black
White
Red
Green
Blue
RRGGBB

Customizing Your Web Site

BorderColor=*color*

where *color* specifies the border color, and can be:

Green

Black

White

Red

Blue

RRGGBB

BorderWidth=*width*

where *width* specifies the width of the border around the image, and can be:

0 No border (the default)

n The number *n* determines the thickness of the border.

Example:

| | | |
|-----------------------|-------------|-------------|
| BorderWidth=1: | 1439 | 1440 |
| BorderWidth=3: | 1441 | 1442 |

Figure 3. Example of the *BorderWidth* option

BorderIndent=*highlighting*

where *highlighting* specifies highlighting for upper and right border edges (3D beveled effect), and can be:

In Upper and right border edges are shaded.

Out Upper and right border edges are lighter.

Example:

| | |
|-----------------------------------|-----------------------------|
| BorderIndent=In (default): | 1234567890: |
| BorderIndent=Out: | The time is 13:12:13 |

Figure 4. Example of the *BorderIndent* option

BorderIndentColor=*color*

where *color* specifies the color for the border edge (3D beveled effect), and can be:

BorderColor

Black

White

Red

Green

Blue

RRGGBB

Customizing Your Web Site

BorderIndentWidth=width

where *width* specifies the width of the border edge (3D beveled effect), and can be:

- 0** No border edge (the default)
- n* The number *n* determines the thickness of the border edge.

Example:

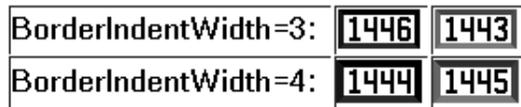


Figure 5. Example of the *BorderIndentWidth* option

FontName=font

where *font* specifies the font used, and can be:

- Block1**
- LCD

FontSize=size

where *size* specifies the font size (width x height), and can be:

- 8x12**
- 7x11
- 9x13
- 10x14

Example:

| | Block1 | LCD |
|-------|---------------|------------|
| 7x11 | 264 | 265 |
| 8x12 | 266 | 267 |
| 9x13 | 268 | 269 |
| 10x14 | 270 | 271 |

Figure 6. Example of fonts and font sizes

Counter option

Format=format

where *format* specifies the format for displaying the *counter_value*, and can be:

- %%d** No padding
- %%nd** Pad with blanks; width=*n*
- %%0nd** Pad with zeros; width=*n*

Customizing Your Web Site

Example:

| | |
|----------------------------------|------------|
| No padding (default): | 268 |
| Pad with 0's (Format=%010d): | 0000000272 |
| Pad with blanks's (Format=%10d): | 273 |

Figure 7. Example of counter format options

Date and time option

Format=strftime()-format

Specifies the format for displaying the *date* and *time*:

- **Default:** http_time format
- *strftime()-format*
 - Use %20 to represent a blank.
 - For all other options, see Table 3 on page 180.

Timebase=time

Specifies the time used:

- **Local**
- GMT (Greenwich mean time)

Example:

| | |
|---|--------------------------------|
| Local time, http_time format (default) | Thu, 16 Jan 1997 13:34:26 EST |
| Local time, Format=lt%20is%20now%20%20%H:%20M:%20S%20on%20%20m/%20d/%20y | It is now 13:34:26 on 01/16/97 |
| GMT time (use option, Timebase=GMT) | Thu, 16 Jan 1997 10:34:27 GMT |

Figure 8. Example of date and time formats

Text to gif option

Text=string

Specifies the text string that will be converted to a gif. Use %20 to represent a blank.

Example:

The following URLs show an example of how you can display text and page count on a Web page:

Customizing Your Web Site

```



```

The URLs in this example display the following information on the Web page:

This Web page has been accessed *n* times since January 1, 1997.

In this example:

- The text is displayed in black (default) on a white background.
- The page count (*n*) is displayed in red on a white background.

Server-side includes and support for text-based browsers

Server-side includes can be used with the apicounter function to display counter values on text-based browsers.

Example:

```
<!--#exec cgi="/cgi-bin/apicounter/counter_name" -->
```

For *counter_name*, enter the name of the counter file to be displayed, for example, **sample.cnt**.

The server-side include returns a text string which can be displayed by either text or graphics-based browsers. HTML tags can also be used to format the result. The graphics options described in “Common options” on page 173 cannot be used. However, the format option described in “Counter option” on page 175 can be used to pad the result with zeros or blanks.

Example:

```
<B><!--#exec cgi="/cgi-bin/apicounter/sample.cnt?Format=%010d" --><B>
```

This example will display the counter file **sample.cnt** using the browser's bold font. The counter will be displayed with leading zeros.

For more information and examples, go to URL <http://your.server.name/counter1.html>.

For a description of server-side includes and server configuration information, refer to “Using server-side includes to insert information into CGI programs and HTML documents” on page 178.

Using server-side includes to insert information into CGI programs and HTML documents

Server-side includes allow you to insert information into CGI programs and HTML documents that the server sends to the client. This section describes the command format for using server-side includes and explains how to use the commands needed to make server-side includes work in your CGI programs and HTML documents.

Considerations for using server-side includes

Before using server-side includes on your server, there are a few issues you should consider. One issue is performance. Performance can be significantly impacted when the server is processing files while sending them. Another issue is security. Letting ordinary users execute commands can be a security risk. Be very careful when deciding which directories you use server-side includes in and which directories you use the `exec` command in. You can minimize the security risk if you do not enable the `exec` command.

You should also note that you cannot reference files recursively. For example, if you are running file `sleepy.html` and the program finds `<!--#include file="sleepy.html" -->` the server doesn't detect the error and the server loops until the server abends. However, you can reference files within files. For example, file `sleepy.html` references file `smiley.html` and file `smiley.html` references `dopey.html`.

Preparing to use server-side includes

To use server-side includes, you need to add the `AddType` directive to your configuration file. Two examples follow:

Examples:

```
AddType .shtml text/x-ssi-html 8 bit 1.0
AddType .htmls text/x-ssi-html 8 bit 1.0
```

Note: If you use file extensions other than `.shtml` or `.htmls`, you should check the `AddType` directive to see if that extension already exists. See the configuration file, appendix listing, or the MIME form for a list of existing `AddType` directives.

You can also use the `imbeds` directive to specify whether server-side includes can be used in HTML documents, CGI programs, or both. Examples of this directive follow:

Customizing Your Web Site

Examples:

```
imbeds value
imbeds on
```

Default: `imbeds on`

For more information about the `imbeds` directive, see “`imbeds` - Specify whether server-side includes will be dynamically imbedded” on page 39.

The server does not process your error files for `imbeds`, regardless of the file extensions or use of the `imbeds` directive.

Format for server-side includes

The current date, the size of a file, the last change of a file are examples of the kind of information that can be sent to the client. There are commands that need to be included in the HTML document comments. The commands have the following format:

Syntax

The following is the syntax format for enabling server-side includes on the server:

```
<!--#directive tag=value ... -->
<!--#directive tag="value" ... -->
```

The quotes around *value* are optional. They are required for imbedding spaces.

Directives for server-side includes

This section explains the directives that are accepted by the server for server-side includes.

config - controls file processing

Use this directive to control certain aspects of file processing. Valid tags are `cmntmsg`, `errmsg`, `sizefmt`, and `timefmt`.

cmntmsg - specify the message appended to the beginning of text: Use this tag to specify the message that gets appended to the beginning of any text that follows a directive specification and comes before `-->`.

Example:

```
<!--#config cmntmsg="[This is a comment]" -->
<!--#echo var=" " extra text -->
```

Customizing Your Web Site

Result: (Output from the echo) <!--This is a comment extra text -->

Default: [the following was extra in the directive]

errmsg - specify the message sent to the client: Use this directive to specify the message that gets sent to the client if an error occurs when a file is being processed. The message gets logged in the server's error log.

Example:

```
<!--#config errmsg="[ An error occurred]" -->
```

Default: "[an error occurred while processing this directive]"

sizefmt - specify file size format: Use this directive to specify the format to be used when the file size is displayed. In the following examples, bytes is the value used for a formatted number of bytes. abbrev is used for displaying the number of kilobytes or megabytes.

Example 1:

```
<!--#config sizefmt = bytes -->
<!--#fsize file=foo.html -->
```

Result: 1024

Example 2:

```
<!--#config sizefmt=abbrev -->
<!--#fsize file=foo.html -->
```

Result: 1K

Default: "abbrev"

timefmt - specify date format: Use this directive to specify the format to be used when providing dates.

Example:

```
<!--#config timefmt="%T %D" -->
<!--#flastmod file=foo.html -->
```

Result: "10/18/95 12:05:33"

Default: "%a, %d %b-%Y %T %Z"

The following strftime() formats are valid with the timefmt tag:

Table 3 (Page 1 of 3). Conversion Specifiers Used by strftime()

| Specifier | Meaning |
|-----------|-----------------|
| %% | Replace with %. |

Customizing Your Web Site

Table 3 (Page 2 of 3). Conversion Specifiers Used by `strftime()`

| Specifier | Meaning |
|-----------------|--|
| %a | Replace with the abbreviated weekday name. |
| %A | Replace with the full weekday name. |
| %b | Replace with the abbreviated month name. |
| %B | Replace with the full month name. |
| %c | Replace with the date and time. |
| %C | Replace with the century number (year divided by 100 and truncated). |
| %d | Replace with the day of the month (01-31). |
| %D | Insert the date as %m/%d/%y. |
| %e | Insert the month of the year as a decimal number (01-12). Under C POSIX only, it is a 2-character, right-justified, blank-filled field. |
| %E[cCxyY] | If the alternative date/time format is not available, the %E descriptors are mapped to their unextended counterparts. For example, %EC is mapped to %C. |
| %Ec | Replace with the alternative date and time representation. |
| %EC | Replace with the name of the base year (period) in the alternative representation. |
| %Ex | Replace with the alternative date representation. |
| %EX | Replace with the alternative time representation. |
| %Ey | Replace with the offset from %EC (year only) in the alternative representation. |
| %EY | Replace with the full alternative year representation. |
| %h | Replace with the abbreviated month name. This is the same as %b. |
| %H | Replace with hour (23-hour clock) as a decimal number (00-23). |
| %I | Replace with hour (12-hour clock) as a decimal number (00-12). |
| %j | Replace with the day of the year (001-366). |
| %m | Replace with the month (01-12) |
| %M | Replace with minute (00-59). |
| %n | Replace with a new line. |
| %O[deHImMSUwWy] | If the alternative date/time format is not available, the %E descriptors are mapped to their unextended counterparts. For example, %Od is mapped to %d. |
| %Od | Replace with the day of the month, using the alternative numeric symbols, filled as needed with leading zeroes if there is any alternative symbol for zero, otherwise with leading spaces. |
| %Oe | Replace with the day of the month, using the alternative numeric symbols, filled as needed with leading spaces. |
| %OH | Replace with the hour (24 hour clock) using the alternative numeric symbols. |
| %OI | Replace with the (12 hour clock) using the alternative numeric symbols. |
| %Om | Replace with the month using the alternative numeric symbols. |
| %OM | Replace with the minutes using the alternative numeric symbols. |
| %OS | Replace with the seconds using the alternative numeric symbols. |

Customizing Your Web Site

Table 3 (Page 3 of 3). Conversion Specifiers Used by strftime()

| Specifier | Meaning |
|-----------|--|
| %OU | Replace with the week number of the year (Sunday as the first day of the week, rules corresponding to %U) using the alternative numeric symbols. |
| %Ow | Replace with the weekday (Sunday=0) using the alternative numeric symbols. |
| %OW | Replace with the week number of the year (Monday as the first day of the week) using the alternative numeric symbols. |
| %Oy | Replace with the year (offset from %C) in the alternative representation and using the alternative numeric symbols. |
| %p | Replace with the local equivalent of AM or PM. |
| %r | Replace with the string equivalent to %l:%M:%S %p |
| %R | Replace with time in 24 hour notation (%H:%M) |
| %S | Replace with seconds (00-61). |
| %t | Replace with a tab. |
| %T | Replace with a string equivalent to %H:%M:%S. |
| %u | Replace with the weekday as a decimal number (1 to 7), with a 1 representing Monday. |
| %U | Replace with the week number of the year (00-53) where Sunday is the first day of the week. |
| %V | Replace with the week number of the year (01-53) where Monday is the first day of the week. |
| %w | Replace with the weekday (0-6) where Sunday is 0. |
| %W | Replace with the week number of the year (00-53) where Monday is the first day of the week. |
| %x | Replace with the appropriate date representation. |
| %X | Replace with the appropriate time representation. |
| %y | Replace with the year with the century. |
| %Y | Replace with the year with the current century. |
| %Z | Replace with the name of the time zone or no characters if the time zone is unknown. |

The operating system configuration determines the full and abbreviated month names and years.

echo - specify environment variables

Use this directive to display the value for specified environment variables using the var tag. If a variable is not found, a **(None)** is displayed. The following environment variables can be displayed:

Customizing Your Web Site

DATE_GMT

The current date and time in Greenwich Mean Time. The formatting of this variable is defined using the `config timefmt` directive.

DATE_LOCAL

The current date and local time. The formatting of this variable is defined using the `config timefmt` directive.

DOCUMENT_NAME

This is the name of the topmost document. If the HTML was generated by a CGI, this will contain the name of the CGI.

DOCUMENT_URI

The full URL the client entered, without the query string.

LAST_MODIFIED

The current date and time that the current document was last modified. The formatting of this variable is defined using the `config timefmt` directive.

QUERY_STRING_UNESCAPED

The search query sent by the client. This is undefined unless HTML was generated by a CGI.

SSI_DIR

The path of the current file, relative to `SSI_ROOT`. If the current file is in `SSI_ROOT`, this value is `"/`.

SSI_FILE

The file name of the current file.

SSI_INCLUDE

The value used in the include command that retrieved this file. This is not defined for the topmost file.

SSI_PARENT

The path and file name of the includer, relative to `SSI_ROOT`.

SSI_ROOT

The path of the topmost file.
All include requests must be in this directory
or a child of this directory.

Example:

```
<!--#echo var=SSI_DIR -->
```

Also, **echo** can display a value set by the **set** or **global** directives.

Customizing Your Web Site

exec - specify CGI programs

Use this directive to include the output of a CGI program. Exec discards any HTTP headers CGI outputs EXCEPT for:

content-type

determines whether to parse the body of the output for other Includes.

content-encoding

determines if translation needs to be done (ebcdic/ascii).

last-modified

replaces the current last modified header value if it is later.

cgi - specify CGI program URL

Use this directive to specify the URL of a virtual path to a CGI program.

Example 1:

```
<!--#exec cgi="/cgi-bin/program/path_info?query_string"-->
```

In the example, program is the cgi program to be executed, path_info are the parameter passed to the program as environment variables, and query_string are the parameters passed to the program as environment variables.

Example 2:

```
<!--#exec cgi="&path;&cgiprogram;&pathinfo;&querystring;"-->
```

Example 2 shows the use of variables.

flastmod - display time and date document was changed

Use this directive to display the last time and date the document was changed. The formatting of this variable is defined using the config timefmt directive. The file and virtual tags are valid with this directive, and the meaning is the same as it is for the include directive.

Directive Formats:

```
<!--#flastmod file="/path/file" -->  
<!--#flastmod virtual="/path/file" -->
```

Example:

```
<!--#flastmod file="F00" extra text -->
```

Customizing Your Web Site

Result: 12May96 <!--This is extra text-->

fsize - display file size

Use this directive to display the size of the specified file. The formatting of this variable is defined using the config `sizefmt` directive. The `file` and `virtual` tags are valid with this directive, and the meaning is the same as it is for the `include` directive.

Examples:

```
<!--#fsize file="/path/file" -->
<!--#fsize virtual="/path/file" -->
```

Result: 1K

global - defines global variables

Use this directive to define global variables that can be echoed later by this file, or any included files.

Example:

```
<!--#global var=VariableName value="Some Value" -->
```

include - includes a document in output

Use this directive to include a document (the text from a document) in the output. The `file` and `virtual` tags are valid with this directive:

file - specify file name: Use this tag to specify the name of a file.

For **`flastmod`**, **`fsize`**, and **`include`**, **`file`** is assumed to be relative to `SSI_ROOT` if preceded by a `/`. Otherwise, it is relative to `SSI_DIR`. The file specified must exist either in `SSI_ROOT` or in one of its descendents.

Example:

```
<!--#include file="/path/file" -->
```

virtual - specify a document URL: Use this tag to specify the URL of a virtual path to a document.

For **`flastmod`**, **`fsize`**, and **`include`**, **`virtual`** is always passed through the server's mapping directives.

Example:

```
<!--#include virtual="/path/file" -->
```

Customizing Your Web Site

set - sets variables to be echoed

Use this directive to set a variable that can be echoed later by only this file.

Example:

```
<!--#set var="Variable 2" value="AnotherValue" -->
```

Variables: Server-side includes also allow you to echo a variable already set. While defining a directive, you can echo a string in the middle of "value." For example:

```
<!--#include file="&filename;"-->
```

If an unrecognized variable is found, nothing gets displayed.

Server-side includes look for the variable, echos where the variable is found, and proceeds with the function. You can have multiple variable references. When server-side includes encounter a variable reference INSIDE a server-side include directive, it attempts to resolve it on the SERVER side. The following example escapes the & so that server-side includes does not recognize it as a variable. In the second line, the variable "&index;" is a server-side variable and is used to construct the variable name "var1". The variable ˆ is a client side variable, so the & is escaped to create the value ":frˆd" or "fred" with a circumflex over the e.

```
<!--#set var="index" value="1" -->
<!--#set var="var&index;" value="fr\&circ;d" -->
<!--#echo var="var1" -->
```

The following characters that can be escaped. Note that escaped variables are preceded with a backslash (\).

| | |
|----|----------------------|
| \a | Alert(bell) |
| \b | Backspace |
| \f | Form feed (new page) |
| \n | New line |
| \r | Carriage return |
| \t | Horizontal tab |
| \v | Vertical tab |
| \' | Single quote mark |
| \" | Double quote mark |
| \? | Question mark |
| \\ | Backslash |
| \- | Hyphen |
| \. | Period |
| \& | Ampersand |

Chapter 6. Rating Web sites and serving rated Web information

Using the Platform for Internet Content Selection (PICS), users of Internet applications, such as the World Wide Web, FTP, and Gopher, can filter the material they encounter and accept or reject the material based on its ratings. This filtering allows parents, businesses, schools, or discerning individuals to block the access to inappropriate and objectionable material. For the most up-to-date PICS information, see the World Wide Web Consortium's PICS Web site (<http://www.w3.org/pub/WWW/PICS/>). The specifications published at this Web site enable:

- Content providers (people who publish information on the Web) to rate and label their own documents. These can be HTML files, or other files that contain images, sound, or animations.
- Independent rating services to rate and label documents published by other Web sites and to distribute the labels to whomever requests them.
- Internet users (browsers and other clients) to request these labels and determine how to handle rated and unrated information.

The Internet Connection Secure Server makes it easy for you to store and serve the rating labels for the documents you publish. It also allows you to act as a rating service or label bureau by providing a means for you to maintain and distribute rating labels for other Web sites.

Who can rate Web sites

Web sites can rate themselves or be rated by a third party, called a **rating service**. A rating service evaluates Web content according to their own published criteria and then distributes the labels through a **label bureau**. Often a rating service acts as its own label bureau and distributes its own labels.

Some rating services will also give you assistance in assessing and labeling your own site and documents. The World Wide Web Consortium publishes a list of PICS self-rating services at <http://www.w3.org/pub/WWW/PICS/selfrat.htm>.

The PICS specification does not determine who can or will act as a rating service. The World Wide Web Consortium publishes a list of PICS third-party rating services at <http://www.w3.org/pub/WWW/PICS/pics.htm>. In addition, anyone who wants to can set up a rating service. You can set up such a service by:

Rating Web Sites and Information

- Deciding on a rating system
- Publishing the rating system
- Rating documents and creating the rating labels
- Establishing a Web site (URL) that clients can access to get your labels

A rating service can choose any criteria on which to rate Web sites. While some might rate Web sites for their violence or sexual content, others could choose to rate educational content, political correctness, or even how "cool" the site is. Also, a rating service can rate any and all Web sites that it wants to rate.

Having your Web site and pages rated is often desirable. In fact, it may even be necessary for your Web site to be rated in order to be viewed by a PICS-enabled client. Understanding how Web clients use the PICS labels and ratings will help make this clear.

How Web clients use PICS

PICS-enabled clients allow the users to determine which rating services they want to use and, for each rating service, which ratings are acceptable and which are unacceptable.

For example, a family might choose a rating service that rates documents according to their sexual content. The rating service might have a low rating for romance, a higher rating for passionate kissing, and yet higher ratings for more explicit sexual activity. The parents might decide that documents containing romance are the highest acceptable rating for their household. They would then configure their browser to reject all documents that are unrated or contain a higher rating from this rating service.

In another example, the Hi-Tek Systems Corporation could label its own documents with a "For Hi-Tek Use Only" and could equip all its employees with browsers configured to accept only documents with that rating.

There are several steps in this process:

The client sends a request

When a PICS-enabled client requests a document, it indicates in the request which rating services are of interest. For example, assume these parents had configured their browser to evaluate rating labels from *The Best* rating service. When their children click a link to an HTML document, the browser request would also ask for the rating labels that were assigned to the document by this rating service.

The server sends a response

Assume the PICS-enabled server has a copy of the labels the client is requesting. When the server receives the client's request, it sends the labels along with the requested document. However, if the server does not

Rating Web Sites and Information

support PICS or does not have copies of labels from that particular rating service, it sends the requested document anyway.

The client checks the server response first

The client first checks to see if the requested ratings labels are imbedded in the document (in the meta information) or if they were sent along with the document. Some clients might accept rating information that is imbedded in the file. Others might require a separate label from a registered rating service and a guarantee that it was created by that service. If the client successfully finds the label information it wanted, it evaluates the rating and either displays the document or blocks it and displays a message.

The client contacts the rating service, if necessary

If the client does not receive the label information with the requested document from the server, it might send a subsequent request directly to the rating service asking for the label information for that document. This requires a second connection, which takes longer and can discourage future visits to that site. The browser waits until the label information is returned before it displays **any** data.

Faster response time is the main reason why rating labels for a site should reside **at the site**.

How the Internet Connection Secure Server helps you manage PICS labels

Whether your Web server publishes Web documents or you are a rating service and want to provide the labels for other Web sites, the Internet Connection Secure Server can help you manage PICS labels.

Note: If you are going to use your server to rate your own documents or to run a label bureau, we strongly suggest that you use the default server port (80).

PICS for Web site administrators

As more browsers are configured to block access to unrated documents, it behooves you to have your Web site rated. And because it saves time when a browser can get the ratings when it sends its initial request, it behooves you to store the ratings for your pages on your own server. With the Internet Connection Secure Server's PICS support, you can manage the labels from one central file and serve them with requested pages and documents. These labels can be:

- Self-assessed according to your own criteria

Rating Web Sites and Information

If you are establishing your own rating service, you can rate your own site according to your published criteria.

- Self-assessed according to the published criteria of a voluntary rating service

Voluntary rating services, such as SafeSurf (<http://www.safesurf.com>) trust Web administrators to be honest in the assessment of their own pages.

- Assessed by a third-party rating service according to the service's criteria

In this case, you might contact the rating service and request that they rate your Web site (if they have not already done so) and send you the label information. In fact, you might want to contact several rating services to have your site rated for different subject criteria. If the third-party rating services have the Internet Connection Secure Server, this process can be simplified with an electronic request. See "How to request PICS label information" on page 195.

Once the ratings are established, Web administrators can do one of three things:

- Manually edit each of their HTML files, inserting the rating information in the headers. See "Managing PICS labels for your Web site in each document."
- Use the label information to create PICS-compliant rating labels, store the labels in their file system, and use the Internet Connection Secure Server's PICS configuration file to manage and transmit them. See "Managing PICS labels for your Web site from a central file."
- Let the system automatically store the transmitted rating labels and update your PICS configuration file for you. This can only be done when electronically requesting labels for a third-party rating service that has the Internet Connection Secure Server. See "How to request PICS label information" on page 195.

Managing PICS labels for your Web site in each document

You can edit each of your HTML files and embed PICS ratings information in the meta element of the document header. This process is entirely manual and therefore time-consuming, error-prone, and difficult to maintain. It does not incorporate any of the security mechanisms (message digest, digital signature, etc.) that would guarantee the validity of the label, if this is important to the requesting client. The PICS specification (<http://www.w3.org/pub/WWW/PICS/>) explains how you can embed rating information in each document. It is not covered here.

Managing PICS labels for your Web site from a central file

The Internet Connection Secure Server's PICS support allows you to store the rating labels for all the documents on your Web site and manage them from a central file. The labels are sent along with your Web pages when a client requests them.

Rating Web Sites and Information

In addition to the rating labels, you must also have a PICS-compliant rating system description file that describes the rating system used to rate your documents. These are called RAT files, and rating services will provide them along with their labels.

Once you have both the labels and the RAT file, you can use the PICS configuration file to manage these labels from a central point. See “How to manage PICS labels from a central file” on page 192.

PICS for rating services and label bureaus

Because many Web sites will want their pages rated, you have an opportunity to provide a service to a large number of Web sites.

- Content providers will contact your organization to request that you rate their Web site and provide them with the labels so that they will be able to serve the labels along with their Web documents themselves.
- Clients will connect to your server electronically to request labels for pages they are attempting to view only when they **cannot** get the label information with the requested pages.

The PICS configuration file provides you with the means to manage the labels for other Web sites and transmit them when requested.

The PICS specifications allow anyone to set up a rating service, define the criteria by which they rate Web sites and documents, and then provide the ratings. With the Internet Connection Secure Server's PICS support, you can establish your server as a rating service and maintain and distribute labels for other Web sites. You can rate documents at a Web site individually or use wildcard characters to quickly assign the same rating to all or part of a Web site's offerings. You will need to create these labels and your own RAT file. The RAT file is a PICS-compliant rating file that describes the rating system used to rate documents. Once you have both the labels and the RAT file, you can use the PICS configuration file to manage these labels from a central point. Your server will then be able to automatically send the rating labels you have assigned when a client requests them. See “How to manage PICS labels from a central file” on page 192.

If a Web site that you have rated requests the labels for their pages, you can also provide them with all their current ratings. Unfortunately, the World Wide Web Consortium has not yet defined a standard for the label bureaus or rating services to send a Web site all their label information. This means that the method for this exchange will have to be determined by the rating services and the Web sites that ask for them.

If the Web sites and the rating service (or label bureau) both have the Internet Connection Secure Server, they can electronically exchange rating labels and label entries for their PICS configuration file. In this case, the rating labels will be automatically stored on the server and the PICS configuration file will be updated so that it can transmit the labels with the requested documents.

Rating Web Sites and Information

If not, we are assuming that the rating services will send a file of all the required label information to the Web site administrators. Once the administrators receive this information, they will use whatever method is available on their server to create PICS rating labels and enable their server to transmit them with the requested documents.

How to manage PICS labels from a central file

Managing PICS labels on the Internet Connection Secure Server requires three things:

1. A rating (RAT) file that describes the ratings

If you are starting your own rating service or label bureau, you will need to create a file that describes your rating system. This file must be in the machine-readable format detailed in the PICS technical specifications and it should have the **.rat** extension. If you are getting your labels from a third-party rating service, you must also get a copy of their RAT file.

2. The rating labels themselves

Whether you are maintaining labels for your own Web site or, as a rating service, maintaining labels for other sites, you will need to store the labels in your server's file system, one label per file. Rating services will rate documents on the Internet and create the rating label files themselves. Web sites will either rate their own site and create the label files or they will request the rating labels from third-party rating services.

If you are getting your rating labels from a third-party rating service that also has the Internet Connection Secure Server, you can request the labels electronically and they will be sent and stored directly on your system for you. Otherwise, you may need to do some editing of the information you receive before creating rating labels to store in your file system.

When creating PICS rating labels, be sure to follow the PICS specification. See "How to create PICS labels" on page 195. We recommend you use **.lbl** for the extension on your label files and we have included a predefined `AddType` directive in the configuration file for this extension.

3. The PICS configuration file

This file provides a mapping between the actual rating labels and the documents they rate. It enables the server to quickly respond to HTTP, FTP, and Gopher requests. If you are getting your rating labels from a third-party rating service that also has the Internet Connection Secure Server, your PICS configuration file will automatically be updated with entries for the labels you receive. If you are a rating service or if you receive rating labels from third-party rating services that have a different server, you will need to maintain the PICS configuration file yourself. You

Rating Web Sites and Information

can use the online Configuration and Administration Forms to update and maintain this file or you can edit it manually. See “How to update the PICS configuration file” on page 196.

Storing the PICS files on your server

You will need to store both the RAT file and rating labels in files on your server.

The RAT file should be available from a rating service's Web site. The rating labels must be stored one label per file.

You can use any directories, subdirectories, and file names that make sense at your site and for your implementation. We recommend that Web sites have a separate directory or subdirectory for each third-party rating service that they use. This is **required** for automatic updates when requesting labels from rating services that have the Internet Connection Secure Server.

Our examples use a file extension of *.lbl* on each rating label file. This is also the extension for any label files the Internet Connection Secure Server transmits electronically.

Managing PICS labels for your own Web site

Follow these steps to store rating labels in your file system and configure your server so it sends these labels when clients request them.

1. Obtain a copy of the RAT file from the rating services you want to use and store it in your file system on your server.
2. If you are getting rating labels from a third-party rating service that has the Internet Connection Secure Server:
 - Use the online Configuration and Administration Forms to request the labels and the entries for your PICS configuration file electronically. When you receive these files, your server will automatically be updated for you. See “How to request PICS label information” on page 195.

If you are **not** getting rating labels from a third-party rating service that has the Internet Connection Secure Server:

- Obtain the ratings from the third-party rating service or rate your own documents.
- Create labels according to the format published in the PICS specification. See “How to create PICS labels” on page 195.
- Store the labels in separate files, one label per file, in your server's file system.
- Tell your server which documents are rated, where the actual rating labels can be found, and which rating service provided the labels. You do this by adding

Rating Web Sites and Information

entries to the PICS configuration file to associate the rated documents with their label files. You can use the online Configuration and Administration Forms to update and maintain this file or you can edit it manually. See “How to update the PICS configuration file” on page 196.

Starting a PICS rating service and label bureau

Follow these steps to configure your server as a PICS rating service, store rating labels for other Web sites, and serve them in response to client requests.

1. Define a rating system and create your own RAT file. Check the World Wide Web Consortium's PICS specification (<http://www.w3.org/pub/WWW/PICS/services.html>) for instructions on how to do this. It includes the syntax for the machine-readable format of the RAT file.
2. Establish two URLs for your rating service. One URL identifies your service by name. Include this URL in your RAT file. The other URL is for label requests. You must direct all the label requests that come to your server to this specific URL.

The PICS specification has no requirements regarding these URLs; you may choose any URL that you like.

Add the `Service` directive to inform the server that you are a PICS service and specify where to direct the PICS rating label requests. For example:

```
Service /Ratings INTERNAL:PICS-Ratings
```

Replace `/Ratings` with the path and file name portion of the URL you will use for label requests. For example, if you publish the URL <http://www.coolratings.com/CoolSite>, you would only include `/CoolSite` in the `Service` directive.

3. Rate documents and Web sites according to your established rating system.
4. Create rating labels for these documents and sites and store them in your server's file system, one label per file. See “How to create PICS labels” on page 195.
5. Tell your server which documents you have rated, what host serves them, and where the labels can be found in your file system. You do this by putting entries in the PICS configuration file that associates the rated documents and their specific label files. You can use the online Configuration and Administration Forms to update and maintain this file or you can edit it manually. See “How to update the PICS configuration file” on page 196.
6. Make the URL you will use for label requests known to the public.
Notify all your subscribers and users to send their requests for rating labels to this URL. PICS-enabled clients and servers will use this URL to contact your server for labels.

How to create PICS labels

In general, a label file is a text file containing a label. Carefully review the format of labels given by the PICS Rating Services and Rating Systems specification (<http://www.w3.org/pub/WWW/PICS/services.html>).

PICS label extensions

The Internet Connection Secure Server has added extensions to this format to save you repetitious data entry and to allow you to add comments.

Comments for your own use

You can insert comments for your own use into label files. Begin these comment lines with '#'. Lines beginning with '#' are not sent to clients. This type of comment is an addition to the "comment" statements used inside labels. "Comment" statements in labels are sent to clients.

Additional variables

The Internet Connection Secure Server has also defined some variables that you can insert in label files.

%%URL%%

The current URL will be substituted for this variable. When the server receives a request for a rating label that contains for %%URL%%, it replaces this variable with the correct for statement before sending the label.

Note: Do not use this variable on generic labels (those that apply to multiple files).

%%SERVICENAME%%

The service name requested will be substituted for this variable. When the server receives a request for a rating label that contains for %%SERVICENAME%%, it replaces this variable with the correct service statement before sending the label.

How to request PICS label information

If a third-party rating service has the Internet Connection Secure Server, you can electronically request rating labels for all the documents on your Web site that the third-party service has rated. As a response to that request, you will receive both rating

Rating Web Sites and Information

labels and PICS configuration file label entries. Both types of information will automatically be stored on your server.

To electronically request rating label and entries for automatic update:

1. From the default home page (Frntpage.html), select **Configuration and Administration Forms**. When prompted, enter the administration user ID and password you have set up.
2. Select **PICS Services Configuration**. This displays the PICS Services Configuration main page.
3. Select **Request Label Entries from Third-Party Rating Service**.

Note: The third-party rating service **must** have the Internet Connection Secure Server for you to use this feature. If not, the request fails.

If the third-party rating service has rated your Web site, it will return both the rating labels and label entries for your PICS configuration file. The rating labels will be stored in the directory you specified on the form. The label entries will automatically be added to your PICS configuration file.

If the third-party rating service has not rated your Web site, it will return a response indicating that it does not have the information you requested.

How to update the PICS configuration file

The Internet Connection Secure Server provides the PICS configuration file for you to manage PICS labels from a central point and serve them when clients request them. You can use the online Configuration and Administration forms to add, modify, and delete the label entries in the PICS configuration file, or you can edit the file and maintain the data manually.

Using the online Configuration and Administration forms

1. From the default home page (Frntpage.html), select **Configuration and Administration Forms**. When prompted, enter the administration user ID and password you have set up.
2. Select **PICS Services Configuration**. This displays the PICS Services Configuration main page.
3. If you are maintaining labels for your own Web site:

Rating Web Sites and Information

- a. Select **Register Third-Party Rating Services** to register the services that have sent you labels and identify their RAT files. With the PICS example files, initially you will have one entry for the The Best rating service, <http://www.coolness.raleigh.ibm.com/ratings/V1.html>, along with its RAT file, coolness.rat.
- b. Select **Maintain PICS Label Entries for Your Web Site** to view, add, modify, or delete the entries that associate specific documents or pages with your rating labels.

If you are maintaining labels for other Web sites:

- a. Ensure that you have your RAT file stored in your file system.
- b. Select **Register Your Own Rating Service** to register the location of your RAT file on your server.
- c. Select **Maintain PICS Label Entries for Other Web Sites** to view, add, modify, or delete the entries that associate specific documents or pages with your rating labels.

Editing the PICS configuration file manually

The exact name and location of the PICS configuration file is *ETC/ICS_PICS.CNF* (where *ETC* represents the directory specified by the ETC environment variable in the CONFIG.SYS configuration file).

The configuration file consists of a list of paragraphs. There are three types of paragraphs.

- **LabelsFor**

Specifies the ratings given by a particular rating service for documents on a given Web server. For example, one LabelsFor paragraph could cover ratings according to the RSAC rating system for documents on the local server, while another paragraph could cover ratings according to the The Best rating system for documents on the local server.

- **DefineService**

Lists local label files associated with a third-party rating service.

- **DefineLBService**

Lists local label files associated with your own label bureau or rating service.

Note: The PICS configuration file associates Web documents with files containing labels. The labels themselves are stored in separate files, not in the PICS configuration file.

Rating Web Sites and Information

PICS configuration file syntax

LabelsFor: The first line of the paragraph consists of the keyword `LabelsFor`, the name of the server on which the rated documents are found, the name of the rating service, and an opening brace. The body of the paragraph specifies labels for sets of documents. Each paragraph ends with a closing brace.

```
LabelsFor servername servicename {  
    /WebPath1/document1    /path/LabelFile1  
    /WebPath2/document2    /path/LabelFile2  
    ...and so on...  
}
```

servername

This can be the keyword `LOCAL` to indicate documents on this server, or it can be a full URL if documents on remote servers are being rated. Only servers acting as label bureaus (rating services) will need to use a hostname other than `LOCAL`. When your server is providing labels for the documents it hosts, you should always use the keyword `LOCAL` for the hostname. Note that you must specify the protocol and hostname without a trailing slash; thus, `http://www.xyz.com` is acceptable as a hostname on a `LabelsFor` line, but `http://www.xyz.com/` is not.

servicename

The full URL where clients will send their label requests.

/WebPath/document

The Web path and name of the document being rated. This is the path a Web client would use when requesting the document. For example, if the `Naughty/Image1.gif` was on the server `www.rated.xyz.com`, then a Web client would request `http://www.rated.xyz.com/Naughty/Image1.gif`.

Note: You can use wildcard characters (*) to rate multiple documents at once. See "Using wildcards in the PICS configuration file" on page 200.

/path/LabelFile

The fully qualified name of the label file in your file system. (This includes the drive name.)

You cannot use wildcard characters in file names.

A special keyword, `NOTLABELED`, can be used in place of a label file name. This indicates that the given file(s) cannot be labeled; it serves as a shorthand way of creating a label file that contains a "not-rated" label. In the example above, a not-rated error message will be returned to any clients who request a rating for the file `/Unknown.html`.

For example, an actual `LabelsFor` paragraph might look like this:

Rating Web Sites and Information

```
LabelsFor LOCAL http://www.rsac.org/ratingsv01.html {  
  /Naughty/Image1.gif d:\www\pics\labels\AdultsOnly.lbl  
  /Clean/*.html       d:\www\pics\labels\AllAges.lbl  
  /Unknown.html      NOTLABELED  
}
```

DefineService: The first line of the paragraph consists of the keyword `DefineService`, the rating service URL, the quoted name of the rating service, the location and name of the service's RAT file, and an opening brace. The body of the paragraph lists the label files associated with this service, specifying each one with the keyword `LABELFILE`. Each paragraph ends with a closing brace.

```
DefineService servicename "name-of-service" ratingfile {  
  LABELFILE /path/LabelFile1 "description"  
  LABELFILE /path/LabelFile2 "description"  
  ...and so on...  
}
```

servicename
The name (URL) of the rating service.

name-of-service
The name (text) of the rating service, in quotes.

ratingfile
The fully qualified name of the service's RAT file in your file system. (This includes the drive name.)

/path/LabelFile
The fully qualified name of the label file in your file system. (This includes the drive name.)

description
A text description of the label, in quotes.

For example, an actual `DefineService` paragraph might look like this:

```
DefineService http://www.abc.org/rate.html "The ABC's of Ratings" d:\www\pics\rat\abc.rat {  
  LABELFILE d:\www\pics\labels\AdultsOnly.lbl "rated XXX"  
  LABELFILE d:\www\pics\labels\AllAges.lbl "rated GGG"  
}
```

DefineLBSERVICE: This paragraph has the same syntax and format as the `DefineService` paragraph. The only difference is that it uses the `DefineLBSERVICE` keyword and the RAT file and labels that it lists are for your own label bureau and rating service.

Using wildcards in the PICS configuration file

You can use an asterisk (*) as a wildcard only in the LabelsFor paragraphs of the PICS configuration file. When using wildcards, remember that the order of entries within a paragraph is important. For each paragraph, the Internet Connection Secure Server breaks the list of rated documents into two parts: those that contain wildcards, and those that do not contain wildcards.

- When the server looks for labels for a document, it will first try to find the document in the "no-wildcards" list. Order is **unimportant** here. Without wildcards, each entry in the list refers to exactly one document.
- If the server cannot find a match in the "no-wildcards" list, it will try to match the document name against the entries that contain wildcards. Order is **important** here. The server tries to match the requested document against the wildcard entries in the order in which they appear in the configuration file and will use the first entry that matches.

For example, if you want an entry that gives /* as the WebPath/document, serving as a catchall for documents that don't have another rating, then make this the last entry in the paragraph.

Chapter 7. Protecting your server

This chapter describes how to protect the data and files on your server and includes the following topics:

| | |
|--|-----|
| Protection methods | 201 |
| Step 1. Activating protection | 203 |
| Step 2. Passing the requests | 203 |
| Step 3. Deciding what type of protection to use | 204 |
| User name and password protection | 204 |
| Address template protection | 205 |
| How the server processes requests | 205 |
| Step 4. Creating protection setups | 207 |
| Identifying the protection setup to requesters | 208 |
| Specifying the type of authentication | 208 |
| Pointing to the password file | 209 |
| Pointing to a server group file | 209 |
| Specifying valid user names, groups, and addresses | 210 |
| Creating protection setups for SSL client authentication | 212 |
| Step 5. Limiting access to individual files | 214 |
| Using server group files | 214 |
| Using Access Control List (ACL) files | 215 |
| Protection example (without SSL client authentication) | 218 |
| Protection example (with SSL client authentication) | 221 |

For information on providing security for server data on the Internet, see Chapter 9, "Making your communications secure" on page 239.

Protection methods

You announce to the world that you want people to come look at the documents on your server. But once you publicize your server to the Internet, you risk attracting unwanted attention to the system on which it runs. Unauthorized people may try to guess passwords, update files, execute files, or tap into confidential data. Part of the attraction of the World Wide Web is its openness. However, the Web is open to both positive use and abuse.

There are several ways that you can protect your system:

- Place a server meant for public access in a network that is separate from your local or internal network.

Protecting Your Server

- Disable Telnet, rlogin, and finger clients on the system that is running the server. In particular, consider disabling Telnet and TN3270.

- Use packet filtering and firewalls.

Packet filtering allows you to define where data can come from and where it can go. You can configure your system to reject certain source/destination combinations.

A firewall provides a way to separate an internal network from a publicly accessible network, such as the Internet. The firewall can be a group of computers or a single computer that acts as a gateway in both directions, regulating and tracking the traffic passing through it. An example of firewall software is the IBM Internet Connection Secured Network Gateway.

You can configure your server as a proxy server through a firewall. This way, internal users can get out to the Internet, but outside Internet travelers cannot get in. Using a proxy server also ensures that any access to the Internet by internal browsers is done anonymously. Because all of the requests to the Internet are made by the proxy server on the part of the internal user, the remote host does not have access to the name of the specific host that is on the inside of the firewall. For information on configuring your server as a caching proxy, see “Running your server as a proxy” on page 11. For information on using your server on a firewall machine, see the documentation for your firewall software.

- Protect access to CGI scripts.

Using CGI scripts on a Web server can create a security exposure. If you use CGI scripts, you need to protect them by controlling who has access to them.

It is possible to write CGI scripts that display all environment variables. At times these variables may include sensitive data such as user IDs and passwords. So you must be careful about displaying environment variables in your CGI scripts and control access to your CGI scripts. Make sure you know what a CGI program does before you make it available on your server.

- Protect user directories on your server.

By allowing local users to have Web pages, you may be creating a security exposure. You need to carefully control access to the pages on local users' machines. You may choose not to allow any local users to have executable CGI scripts. For more information, see “User directories” on page 52.

The following sections describe how you control who has access to the various files you keep on your server. The steps first tell you how to set up protection by editing the configuration file. When appropriate, the steps tell you which Configuration and Administration forms you can use to perform the same task.

Step 1. Activating protection

The first step to controlling access to your server's resources is to activate protection. You activate protection based on the content of requests that clients send to your server.

This document refers to a *request* as the part of a full URL that follows your server host name. For example, if your server is named fine.feathers.com and a requester enters the following URL on a browser:

http://fine.feathers.com/waterfowl/schedule.html

The request your server receives is: /waterfowl/schedule.html

You can use Protect directives to specify which requests should activate protection. Each Protect directive has a request template. The server activates protection when it receives a request that matches a request template on a Protect directive. The Protect directive also either identifies or contains the protection setup to be used.

See "Access control - Set up access control for the server" on page 76 for details on how to use Protect directives.

You can also use the Configuration and Administration forms to specify which requests should activate protection. From the Configuration and Administration forms page, click **Document protection**.

Step 2. Passing the requests

Besides activating protection for the requests, you must also tell your server which requests to accept for processing.

Use Pass and Exec directives to specify which requests you want your server to accept. The Pass and Exec directives map requests to actual directories and files on your server (the information you are protecting).

Like the Protect directive, each Pass and Exec directive contains a request template. After the server checks to see if a request activates protection, it goes through its list of Pass and Exec directives to determine if it should accept the request. If the request matches a request template on a Pass or Exec directive, the server **accepts** the request. If protection was activated, the server uses the protection setup to determine whether it should **complete** the request.

Protecting Your Server

So for protection to work properly, you must ensure that your Pass and Exec directives accept the requests that your Protect directives activate protection for. Use Pass directives to accept document requests. Use Exec directives to accept CGI program requests.

See “Resource mapping - Redirect URLs” on page 101 for details on how to use the Pass and Exec directives.

Attention: You must put your Protect directives before any Pass or Exec directives in your configuration file.

You can also use the Configuration and Administration forms to specify which requests to accept and how to map them to your actual resources. From the Configuration and Administration forms page, click **Request Routing**.

Step 3. Deciding what type of protection to use

You can use two types of protection to control access to your resources through the server:

- User name and password protection
- Address template protection.

You can use one type of protection by itself or both together. You can use either or both types of protection in protection setups and Access Control List (ACL) files. It is helpful to understand these two different types of protection before you create your protection setups or ACL files.

The following two sections describe each type of protection. Following those two sections is a section describing how the server processes requests based on the type of protection being used.

User name and password protection

With this type of protection, you create user names that you want requesters to use to access your protected resources. You define each user name and assign it a password in a password file.

After you define a user name, you can use it within protection setups and ACL files to specify which user names are valid for different types of requests.

Protecting Your Server

When the server receives a request that activates this type of protection, the server prompts the requester for a user name and password. In order for the request to complete, the requester must return a user name and password that meet the following criteria:

- The user name must be defined in the protection setup or ACL file as valid for the type of request being made.
- The password must match the one defined for the user name in the password file.

Address template protection

With this type of protection, you use address templates to specify valid requester addresses for the different types of requests. You can use address templates in protection setups and ACL files.

When the server receives a request that activates this type of protection, it compares the address of the requester to the templates to determine if the request comes from a valid address. The server can use either the IP address of the requester (for example, 9.67.97.103) or the host name of the requester (for example, any.host.name.com) when comparing against the templates.

Note: In order to compare the requester host names against address templates, you must set the DNS-Lookup directive to `0n`. If the DNS-Lookup directive is set to `0ff` (the default), your server can compare only the IP address of the requester to the address templates. See “DNS-Lookup - Specify whether you want to look up host names of clients” on page 38.

How the server processes requests

Following is a description of 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 again in more detail after going through the steps for creating protection setups.

- 1** Based on the HTTP method of the request, the server refers to the appropriate mask subdirective (DeleteMask, GetMask, Mask, PostMask, or PutMask) 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 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

Protecting Your Server

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 that is on the mask subdirective.

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

If there is not a 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 not a 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 password file that the protection setup points to.

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

Note: It is important to note that the password file 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 password file. The names themselves do not have any relation to the addresses of the requesters.

If there is not a 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 password file. Each user name in the password file has one valid password.

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

If there is not a match, the process ends and the server returns a message to the requester saying that authorization failed.

Step 4. Creating protection setups

When a Protect directive activates protection for a request, it also either identifies the protection setup to use or defines the protection setup as part of the directive. A **protection setup** is a group of protection subdirectives. The subdirectives work together to define how the server should control access to the resources being protected. You can create protection setups three different ways:

- You can create protection setups within the configuration file as part of Protection, Protect, or DefProt directives.

When you create a protection setup on a Protection directive, you give the setup a label that you can point to later from Protect and DefProt directives.

When you create a protection setup on a DefProt or Protect directive, the protection setup is used only for that directive. The setup cannot be pointed to by other DefProt or Protect directives. This type of protection setup is called an **in-line** protection setup.

You indicate you are including a protection setup as part of a Protection, Protect, or DefProt directive by making the last character on the line that contains the directive a left brace character ({}). On each following line you put one protection subdirective and its value. You indicate the end of the protection setup by putting a right brace character (}) by itself on the line following the last protection subdirective. You cannot use comments within the protection setup.

See “Access control - Set up access control for the server” on page 76 for more information and examples of creating protection setups in the configuration file.

- You can create separate protection setup files that you can then point to from Protect and DefProt directives. A protection setup file is a plain text file. Within the file, each line contains one protection subdirective and the value for that subdirective.
- You can use the Configuration and Administration forms to create protection setups within the configuration file. From the Configuration and Administration forms page, click **Document protection** and add, change, or delete either an in-line protection setup or a named protection setup.

Within the protection setup, the protection subdirectives control access to the directory or files being protected. The following sections describe how to use each of the protection subdirectives.

Protecting Your Server

Identifying the protection setup to requesters

For user name and password protection, use the ServerID subdirective to specify a name you want to use to identify the protection setup to requesters. The name does not need to be a real machine name.

When the server sends a requester a prompt for user name and password, it also includes the name you specify on ServerID. Most browsers display this name with the prompt. Because different protection setups can use different password files, having a name associated with the protection setup can help the requester decide which user name and password to send back. Many browsers also attempt to automatically send a user name and password if the requester has previously responded to a prompt from a protection setup with the same name.

Because some browsers such as NetScape cache userid/password by security realm (ServerId) within host, we suggest you follow these guidelines when specifying ServerId and password files in your protection setups:

- Protection setups that use the same password file should use the same ServerId.
- Protection setups that use different password files should use different ServerIds.

If the protection setup is using address template protection only, you do not need to use the ServerID subdirective.

If you are using the Configuration and Administration forms, you specify this name in the **Server identifier** field on the Protection Setup form.

Example:

```
ServerID restricted
```

Specifying the type of authentication

The AuthType subdirective specifies the type of authentication to use when a client sends a password to the server. For user name and password protection, you must use the AuthType subdirective with a value of Basic. With basic authentication, passwords are sent to the server as plain text. They are encoded, but not encrypted.

If the protection setup is using address template protection only, you do not need to use the AuthType subdirective.

Example:

```
AuthType Basic
```

Pointing to the password file

For user name and password protection, use the `PasswdFile` subdirective to specify the path and name of the password file that you want the protection setup to use.

Each password file contains a list of user names and passwords. Each user name has one valid password defined for it. The requester must send back a user name and password that exactly matches a user name and password in the password file.

Because some browsers such as NetScape cache `userid/password` by security realm (`ServerId`) within host, we suggest you follow these guidelines when specifying `ServerId` and password files in your protection setups:

- Protection setups that use the same password file should use the same `ServerId`.
- Protection setups that use different password files should use different `ServerIds`.

If the protection setup is using address template protection only, you do not need to use the `PasswdFile` subdirective.

If you are using the Configuration and Administration forms, you specify the file path and name of the password file in the **Password file** field on the Protection Setup form.

You create and maintain password files using the `htadm` command. See “`htadm` command” on page 293 for information on how to create and maintain password files.

Note: The user names in the password file do not have any relation to the addresses of the requesters. You make up the user names and passwords.

Example:

```
PasswdFile c:\WWW\restrict.password
```

Pointing to a server group file

If you want to use group names in the protection setup, use the `GroupFile` subdirective to specify the path and file name of the server group file that contains the group definitions you want to use. The groups defined within the server group file can then be used by:

- Any mask subdirectives that are part of the protection setup. (The mask subdirectives are `DeleteMask`, `GetMask`, `Mask`, `PostMask`, and `PutMask`.)
- Any ACL file on a directory that is protected by the protection setup.

See “Using server group files” on page 214 more information about server group files.

If you are using the Configuration and Administration forms, you specify the file path and name in the **Group file** field on the Protection Setup form.

Protecting Your Server

Example:

```
GroupFile d:\docs\WWW\restrict.group
```

Specifying valid user names, groups, and addresses

Use the mask subdirectives to specify valid user names, groups, and address templates for different types of requests. The mask subdirectives protect the entire directory that the request is mapped to.

Each request to your server contains an HTTP method field that identifies the type of request being made. See “Methods - Set method acceptance” on page 118 for a description of the HTTP methods supported by the server. Choose which mask subdirectives to use based on the types of requests you want to authorize. For a protection setup to be valid, it must contain at least one of the following mask subdirectives:

- DeleteMask - to authorize DELETE requests.
- GetMask - to authorize GET requests.
- PostMask - to authorize POST requests. (Most HTML forms use the POST method.)
- PutMask - to authorize PUT requests.
- Mask - to authorize requests using any enabled methods not covered by the other mask subdirectives. Other mask subdirectives take precedence over the Mask subdirective if both are present in the protection setup. For example, if a protection setup contains a DeleteMask subdirective and a Mask subdirective, DELETE requests are covered by the DeleteMask subdirective and all other requests are covered by the Mask subdirective.

Rules for specifying user names, group names, and address templates

Following are explanations and examples of the different ways you can specify user names, group names, and address templates on mask subdirectives. The same rules also apply for specifying user names, group names, and address templates in server group files and ACL files. If you are using the Configuration and Administration forms, the same rules apply to the values you can enter in the Get, Put, Post, Delete, and All fields on the Protection Setup form.

- You can specify a user name without an address template. The user name must be defined in the protection setup password file. If the requester returns the user name with the correct password, the server completes the request.

```
GetMask swandude
```

Protecting Your Server

- You can specify an address template without a user name. If the requester address matches the address template, the server completes the request without prompting the requester for a user name and password.

The address template can be based on either IP address or host name. Use the asterisk character (*) as a wildcard in any part of the template. To indicate you want to use address template protection only, precede the template with one of the following: @, Anybody@, Anyone@, or Anonymous@.

```
GetMask    Anybody@123.45.2.*
PostMask   @96.*.*.*
DeleteMask Anonymous@walden.pond.*.*
```

Note: In order to compare the requester host names against address templates, you must set the DNS-Lookup directive to 0n. If the DNS-Lookup directive is set to 0ff (the default), your server can compare only the IP address of the requester to the address templates. See “DNS-Lookup - Specify whether you want to look up host names of clients” on page 38.

- You can specify a user name with an address template. The user name must be defined in the protection setup password file. Separate the user name from the address template with the at sign character (@). If the requester returns the user name with the correct password and the requester address matches the address template, the server completes the request.

```
GetMask    webfoot@96.96.*.*
PostMask   billface@*.ibm.com
```

- You can use the value All or Users with or without an address template to represent all user names defined in the password file. If the requester returns any user name and password defined in the protection setup password file, and the requester address matches any address templates, the server completes the request.

```
GetMask    All
PostMask   All@(96.*.*.*,*.*.ibm.com,123.45.2.*)
```

- You can specify a group name that is defined in the server group file specified on the GroupFile subdirective (see “Using server group files” on page 214).

A group name can include user names, other group names, and address templates in any of the same formats allowed on masking subdirectives. To be valid, any user names included in the group name must also be defined in the protection setup password file. If the requester returns a valid user name and password and any address templates are matched, the server completes the request.

```
GetMask    ducks
PostMask   geese
```

- You can use the values Anybody, Anyone, or Anonymous without an address template or with an address template of (*) to indicate you do not want to use any protection for requests covered by the subdirective. The server completes requests without prompting for a user name and password and without checking the address of the requester.

Protecting Your Server

```
GetMask Anybody
PutMask Anyone@(*)
```

- You can specify multiple items on each subdirective. Separate each item with a comma. The comma is treated as a logical or.

```
GetMask swandude@96.96.*.*,geese,ducks,Anyone@walden.pond.*.*
```

- You can continue a list of user and group names onto a new line by ending the previous line with a comma.

```
GetMask swandude@96.96.*.*,webfoot@water.fowl.com,
geese,billface
```

- You can use parentheses to keep user names and group names together or address templates together.

```
GetMask (gooduser,webfoot)@96.96.*.*,
billface@(water.fowl.com,123.45.2.*),
(gooseegg,bagel,eInada)@(98.*,146.*)
```

If you are using the Configuration and Administration forms, you specify user names, groups and address templates in the **Delete mask**, **Get mask**, **Post mask**, and **Put mask** fields on the Protection Setup form. You use the same rules as explained above.

Creating protection setups for SSL client authentication

Chapter 10, “Using Secure Sockets Layer (SSL)” on page 255 describes how to use Secure Sockets Layer (SSL) with a secure server. If you set up your server for SSL client authentication, the server requests a certificate from any client making an **https** request. The server establishes a secure connection whether or not the client has a valid certificate.

You can restrict who can access documents by using password files and/or user or group authentication in protection setups as described in Chapter 7, “Protecting your server” on page 201. You can further restrict who can access documents by coding SSL client authentication parameters on protection setups, ACL files, or both. (Coding SSL client authentication parameters on protection setups is described in this topic; coding them in ACL files is described in “Step 5. Limiting access to individual files” on page 214.)

Using SSL client authentication parameters as subdirectives, you can specify that the client certificate is valid or you can specify all or part of the Distinguished Name (DN) of a client or of the certification authority (CA) who issued the client's certificate.

When you use SSL client authentication parameters, the server first checks to see if the client certificate is valid, as is. If not, it compares any DN information in a protection setup and then compares any DN information in an ACL file with the DN information in the client's certificate. If the DN information matches, the server serves the document.

Protecting Your Server

You can use the Protection Setup Configuration and Administration form to specify the SSL client authentication parameters or you can specify the following on the Protection or Protect directive:

- The validity of the client certificate.
 - `SSL_ClientAuth` client - indicates that the client certificate is valid without verifying any of the Distinguished Name information in the client certificate. Only the keyword *client* is valid with this parameter.
- All or any of the following parameters that make up a client's Distinguished Name in the client's certificate:
 - `CommonName` - the client's common name
 - `Country` - the country in which the client resides
 - `Locality` - the locality in which the client resides
 - `StateOrProvince` - the state or province in which the client resides
 - `Organization` - the organization of the client
 - `OrgUnit` - the organizational unit of the client
- All or any of the following subdirectives that make up the CA's Distinguished Name in the client's certificate:
 - `IssuerCommonName` - the CA's common name
 - `IssuerCountry` - the country in which the CA resides
 - `IssuerLocality` - the locality in which the CA resides
 - `IssuerStateOrProvince` - the state or province in which the CA resides
 - `IssuerOrganization` - the organization of the CA
 - `IssuerOrgUnit` - the organizational unit of the CA

Examples:

```
Protect /topsecret/* {  
  CommonName "Dr Sheila A. Jones"  
  Organization "RTP Quick Care Center"  
  Mask Anybody@(*)  
}
```

In the above example of an inline Protect directive, SSL client authentication must be set up, and the client must be making an **https** request and have a certificate with a common name of Dr Sheila A. Jones and an organization of RTP Quick Care Center to access the document.

Hints and tips for coding SSL client authentication parameters for protection setups

- Specify any or all of a client or CA's DN.
- Enclose DN information that contains blanks or special characters in double quotes (as shown in the above example).

Protecting Your Server

- Make sure the DN information matches the DN information in the client's certificate. This information is case sensitive and must have the same punctuation.
- Do not use wildcard characters for any of the parameters.

Step 5. Limiting access to individual files

Perform this step only if you want to limit access to specific files on directories already protected by the protection setup.

To limit access to specific files on a protected directory, you create an Access Control List (ACL) file and place it on the directory. The ACL file must be named `.www_acl`.

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 you want all control to come from the ACL file, use the `ACLOverride` subdirective with a value of `0n` in the protection setup. This causes the mask subdirectives in the protection setup to be ignored. All access control is passed to the ACL file.

See “Using Access Control List (ACL) files” on page 215 for more information about ACL files.

If you are using the Configuration and Administration forms, you can specify that you want the protection setup to give all control to ACL files by checking the **Allow ACL file to override masks** box on the Protection Setup form.

Using server group files

You can use group files to classify users into groups.

Protection setups can point to a server group file. The protection setup can then use the groups defined in the server group file on mask subdirectives. If a protected directory contains an ACL file, the rules in the ACL file can also use the groups defined in the server group file.

You can create as many server group files as you need. Create each in a separate text file. Within the server group file, each line contains a group definition using the following format:

```
groupname : user1[,user2[,user3...]]
```

Protecting Your Server

groupname

Any name you want to use to identify the group you are defining. This name can be used on:

- Mask subdirectives within protection setups that point to the server group file. (The mask subdirectives are DeleteMask, GetMask, Mask, PostMask, and PutMask.)
- Access rules within ACL files on directories that are protected with a protection setup that points to the server group file.
- Subsequent group definitions within the same server group file.

user1[,user2[,user3...]]

This can actually be any combination of user names, group names, and address templates. Separate each item with a comma.

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

Generally, the items you specify have to follow the same rules described under “Rules for specifying user names, group names, and address templates” on page 210.

Examples:

```
ducks : (webfoot,billface)@96.96.3.1,swandude
geese : goosegg,bagel@(walden.pond.*.*,123.*.*)
flock : ducks,geese
webbed : All@water.fowl.*
```

In the above example, notice that once the groups named ducks and geese are defined, they can be included as part of the group named flock.

Using Access Control List (ACL) files

This section describes how to “hand code” ACL files. You can also create ACL files using the Configuration and Administration forms. From the Configuration and Administration forms page, click **Access Control Lists**.

You can use ACL files to limit access to specific files on a protected directory.

Each protected directory can have only one ACL file. The ACL file must be named `.www_acl` and be present on the protected directory.

Normally, the mask subdirectives in the protection setup define the first level of access control and then the ACL file further limits access to individual files. However, if you

Protecting Your Server

want all control to come from the ACL file, use the `ACLOverride` subdirective with a value of `On` in the protection setup. This causes the mask subdirectives in the protection setup to be ignored when a protected directory contains an ACL file.

Within the ACL file, each line contains a rule limiting access based on file name, HTTP method, and authorized users, groups, or addresses. The server processes the rules in the ACL file from top to bottom. The server compares the three elements of each rule to the request until a match is found or until the end of the file is reached.

The format for rules in ACL files is:

```
file : method : [SSLauth,] user
```

file A file name or a template for the files you want to protect with this rule. Each file template can contain one asterisk (*) wildcard character.

The server denies any requests for files that are not listed or covered by a template.

method or *method1*[,*method2*],*method3*...]

An HTTP method or list of methods, separated by commas. The methods define what kind of requests the authorized users are allowed to make for the protected files. Any method you specify must also be enabled. See "Methods - Set method acceptance" on page 118 for an explanation of the HTTP methods supported by the server and how to enable them.

SSLauth

If you have implemented SSL client authentication, you can code ACL files that specify any or all parts of the Distinguished Name of a client or the CA who issued the client's certificate. You can also include more than one client or CA parameter in a given ACL. The server requests the client's certificate. The server compares the DN information in the certificate first to DN information in protection setups and then to DN information in ACL files. If the DN information matches, the server returns the document.

The ACL can specify any of all of the following parts of the

- Client's Distinguished Name in the client's certificate:
 - !CommonName - the client's common name
 - !Country - the country in which the client resides
 - !Locality - the locality in which the client resides
 - !StateOrProvince - the state or province in which the client resides
 - !Organization - the organization of the client
 - !OrgUnit - the organizational unit of the client
- CA's Distinguished Name in the client's certificate:
 - !IssuerCommonName - the CA's common name
 - !IssuerCountry - the country in which the CA resides
 - !IssuerLocality - the locality in which the CA resides
 - !IssuerStateOrProvince - the state or province in which the CA resides
 - !IssuerOrganization - the organization of the CA
 - !IssuerOrgUnit - the organizational unit of the CA

Protecting Your Server

user or *user1* [, *user2* [, *user3* ...]]

This can actually be any combination of user names, group names, and IP address templates. Separate each item with a comma.

For user names to be valid, they must be defined in the password file that the protection setup points to. Group names must be defined in the server group file the protection setup points to.

The items you specify have to follow the same rules as you use to specify user names, group names, and address templates on mask subdirectives. See "Rules for specifying user names, group names, and address templates" on page 210.

Examples:

```
*           : GET           : All
*.html      : GET,POST      : geese
golden.*    : GET,POST      : geese,@bean.stalk.*
welcome.html : GET,POST,DELETE : webfoot@123.34.14.2
```

In the above example, any valid user name and password can be used to GET any file on the directory. User names defined for the group geese can be used to POST to any HTML files. From IP address 123.34.14.2, the user name webfoot can be used to DELETE the welcome.html file.

Example

```
patientA.html : GET : !CommonName="Dr Shelia A. Jones"@9.67.*.*
```

In the above example, SSL client authentication must be set up, and the client must be making an **https** request from an IP address beginning with 9.67 and have a client certificate with a common name of Dr Shelia A. Jones to access the patientA.html document.

```
patientA.html : GET : !CommonName=("Dr Shelia A. Jones","Dr Harry S. Smith")@9.67.*.*
```

In the above example, SSL client authentication must be set up, and the client must be making an **https** request from an IP address beginning with 9.67 and have a client certificate with a common name of Dr Shelia A. Jones or Dr Harry S. Smith to access the patientA.html document.

Hints and tips for coding SSL client authentication parameters for ACLs

- Specify any or all of a client or CA's DN.
- Specify the DN information for multiple clients or CAs on a given parameter. The DN information should be separated by a comma and enclosed in parentheses (as shown in the above example).
- Enclose DN information that contains blanks or special characters in double quotes (as shown in the above example).
- Make sure the DN information matches the DN information in the client's certificate. This information is case sensitive and must have the same punctuation.

Protecting Your Server

- All of the ACL rule must be on one line.
- Do not use wildcard characters for any of the parameters.

Protection example (without SSL client authentication)

Following are examples of the files you can use for protecting your server resources. These files do not use any of the secure network communications functions. In the files where they are allowed, comments begin with the pound sign (#) in the first column.

Following is an example of the protection related portion of a server's configuration file.

```
Protection    POND-PROT {
  ServerID    Feathered
  Authtype    Basic
  PasswdFile  C:\tcPIP\etc\flyIng.pwd
  GroupFile   C:\tcPIP\etc\nesters.group
  GetMask     All@*.swimmer.org
  PutMask     quacks,billface
  DeleteMask  bigbird@pond.hq.swimmer.org
}
#
# The above Protection directive defines a protection setup named
# POND-PROT. The Protection directive must be placed before any
# Protect or DefProt directives that point to it. The Mask
# subdirectives restrict access as follows:
# o Any requester from a host name ending with .swimmer.org can
#   GET files. They must be able to enter any user name and
#   password defined in the C:\TCPIP\etc\flyIng.pwd password file.
# o To PUT files, the requester must match the restrictions
#   defined for the group named quacks in the
#   C:\TCPIP\etc\nesters.group server group file.
#   Or, the requester must enter the user name billface and the
#   password defined for it in the C:\TCPIP\etc\flyIng.pwd
#   password file.
# o To DELETE files, the requester must be from the host named
#   pond.hq.swimmer.org. The requester must enter the user name
#   bigbird and the password defined for it in the
#   C:\TCPIP\etc\flyIng.pwd password file.
#
Protect /wetland/creatures/* POND-PROT
#
# Any request beginning with /wetland/creatures/
# activates protection as defined in the protection setup
# labeled POND-PROT
#
```

Protecting Your Server

```
Protect /vegetation/*      C:\WWW\proset\webster.setup
#
# Any request beginning with /vegetation/ activates protection
# defined in the separate protection setup file named
# C:\WWW\proset\webster.setup
#
Protect /flocks/* {
  DeleteMask @9.67.84.5
  MASK       Anybody@9.67.*.*
}
#
# The above Protect directive contains an in-line protection setup.
# Any request beginning with /flocks/ activates protection
# defined as part of the Protect directive. The mask subdirectives
# restrict access as follows:
# o To delete files, the requester must be at the host with
#   IP address 9.67.84.5
# o To use any other enabled HTTP method, the requester must
#   be at a host with an IP address beginning with 9.67.
# (@ by itself and Anybody@ both mean the same thing.)
# Since only address protection is being used, there is no need for
# the Authtype, ServerID, PasswdFile, and GroupFile subdirectives.
#
Pass /wetland/creatures/*  D:\freshwater\animals\*
#
# For requests beginning with /wetland/creatures/, the server
# goes to the d:\freshwater\animals\ directory to find the file.
#
Pass /* D:\SServer\HTML\*
#
# For any requests not matching other Pass directives, the server
# goes to the D:\SServer\HTML\ directory (the document root
# directory). For example, if the server received the request
# /vegetation/cattail.html, it would look for the cattail.html
# file on D:\SServer\HTML\vegetation.
#
```

Following is an example of what the c:\WWW\proset\webster.setup file might look like:

Protecting Your Server

```
ServerID    Webby
Authtype    Basic
PasswdFile  C:\TCPIP\etc\flying.pwd
GroupFile   C:\TCPIP\etc\nesters.group
GetMask     All@(123.45.*,water.*)
PutMask     goosegg,webfooter,nest
DeleteMask  (swandude,billface)@(water.fowl.com,123.45.2.*)
ACLOverride On
#
# The mask subdirectives restrict access based on a combination of
# user names and passwords and address templates. However, since
# ACLOverride is on, these subdirectives are ignored and the ACL
# file is used. For user name and password protection, the ACL file
# must use the C:\TCPIP\etc\flying.pwd password file and the
# C:\TCPIP\etc\nesters.group server group file.
#
```

Following is an example of what the C:\TCPIP\etc\flying.pwd file might look like:

```
bigbird:vfzD1IeUzFzCt:The big guy
duckman:PVz1Y6YFI8IY3:The duck
swandude:FexekmFhY7q8:Swanee
billface:ZereWePF0YJqK:Orange nosed one
goosegg:dh*ySyZJP0qrw:Shut out
```

Note: When you look at the contents of the password file, you cannot see the passwords because they are encrypted. To manage the password file you must use the `htadm` command. Do not edit password files with a text editor.

Following is an example of what the C:\TCPIP\etc\nesters.group file might look like:

```
ducks : (webfoot,billface)@96.96.3.1,swandude
geese : goosegg,bagel@(walden.pond.*,123.*.*)
quacks : ducks,geese,Anybody@43.234.*
nest : All@(water.fowl.*,nesting.*.*)
#
# The quacks group demonstrates how you can use previously
# defined groups in a subsequent group definition. The quacks
# group includes both the ducks and geese groups. Additionally,
# the quacks group includes address template protection for any
# requests from hosts with an IP address beginning with 43.234.
# Requests from matching hosts would not be prompted for a user
# name and password.
```

Following is an example of what the `.www_acl` file on D:\SServer\HTML\vegetation might look like.

Protecting Your Server

```
*           : GET           : All
*.html,     : GET,POST      : nest
billed.html : GET,POST      : duckman,geese
welcome.html : GET,POST,DELETE : bigbird
#
# All restrictions come from the ACL file because the protection
# setup specified ACLOverride On. The user names and groups used
# in the ACL file must be defined in the password and server group
# file identified in the protection setup.
```

Protection example (with SSL client authentication)

You can restrict who can access documents by using password files and/or user or group authentication in protection setups as shown in the examples in “Protection example (without SSL client authentication)” on page 218. You can further restrict who can access documents by coding SSL client authentication parameters in protection setups, ACL files, or both.

The following examples assume that SSL client authentication has been set up by specifying SSL client authentication on the Security Configuration form or by coding the SSLClientAuth directive in the configuration file.

```
|
| Protection SSL_CLIENT_AUTH {
|     CommonName "Dr Sheila A. Jones"
|     Country US
|     Organization "RTP Quick Care"
|     Mask Anybody@(*)
| }
|
```

For a server to serve the document, the server authenticates the client by requesting and ensuring the client has a valid certificate. In this case, for the certificate to be valid, the Distinguished Name of the client must be made up of a Common Name of Dr Sheila A. Jones, an organization of RTP Quick Care Center, and a country of US.

If you want to allow all persons with an organization unit of Pediatrics to access the documents protected by the SSL_CLIENT_AUTH protection setup, you could specify:

```
|
| Protection SSL_CLIENT_AUTH {
|     OrgUnit=Pediatrics
| }
|
```

The Protect directive that maps the request to the protection setup follows the protection setup:

```
Protect \secure_docs\* SSL_CLIENT_AUTH
```

Any request beginning with `secure_docs` activates protection as defined in the protection setup `SSL_CLIENT_AUTH`.

Note: If you code SSL client authentication parameters in ACL files and not in protection setups, you must also have a `Protect` statement like the one in the above example.

Chapter 8. Managing your Web server

Your servers present your company to the world on the Internet and are an integral part of internal operations. To ensure continuous, secure, and efficient operation, Internet Connection Secure Server provides facilities for monitoring and managing your server. This chapter describes how to manage and monitor your servers and includes the following topics:

| | |
|---|-----|
| Simple Network Management Protocol | 223 |
| SNMP commands and protocol | 224 |
| Object IDs and variable names for the Internet Connection Secure Server MIB | 224 |
| Creating an e-mail address to receive SNMP problem reports | 232 |
| Enabling SNMP on your OS/2 system | 232 |
| Providing a security password for SNMP | 233 |
| Enabling and disabling SNMP support | 233 |
| Turning the SNMP support on and off from the httpd command | 234 |
| Monitoring server performance and status | 234 |
| Using the Server Activity Monitor function | 234 |

Simple Network Management Protocol

A network management system is a program that runs continuously and is used to monitor, reflect status of, and control a network. Simple Network Management Protocol (SNMP), a popular protocol for communicating with devices in a network, is the current network management standard. The network devices typically have an SNMP **agent** and one or more subagents. The SNMP agent talks to the **network management station** or responds to command line SNMP requests. The SNMP **subagent** retrieves and updates data and gives that data to the SNMP agent to communicate back to the requester.

The Internet Connection Secure Server provides an SNMP **management information base (MIB)** and SNMP subagent so you can use any network management system, such as IBM NetView for AIX, TME10 Distributed Monitoring, or HP OpenView, to monitor your server's health, throughput, and activity. The MIB data describes the Web server being managed, reflects current and recent server status, and provides server performance data.

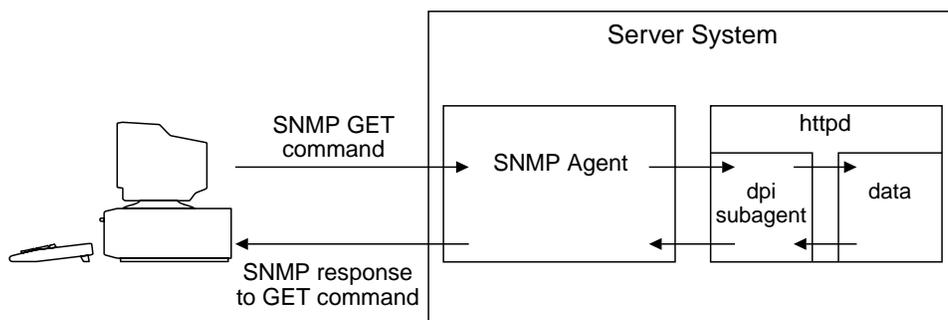
The network management system uses SNMP GET commands to look at MIB values on other machines. It then can notify you if specified threshold values are exceeded. You can affect server performance, by modifying configuration data for a server, to proactively tune or fix server problems before they become server outages.

Managing Your Web Server

SNMP commands and protocol

The system usually provides an SNMP agent for each network management station. The user or a programmer sends a GET command to the SNMP agent. In turn, this SNMP agent sends a GET command to retrieve the specified MIB variable values from a subagent responsible for those MIB variables.

The Internet Connection Secure Server provides a subagent that updates and retrieves MIB data. The subagent responds with the appropriate MIB data when the SNMP agent sends a GET command. The SNMP agent communicates the data to the network management station. The network management station can notify you if specified threshold values are exceeded.



The Internet Connection Secure Server SNMP support includes an SNMP subagent that uses Distributed Protocol Interface (DPI) capability. DPI is an interface between an SNMP agent and its subagents.

Object IDs and variable names for the Internet Connection Secure Server MIB

The Internet Connection Secure Server MIB is modeled after the HTTP MIB proposal. MIB layout includes Variable Name, Object ID, Type, and Description.

The following Variable Names and Object IDs are provided for SNMP support with the Internet Connection Secure Server:

EntityDescription

| | |
|---------------|---|
| Description | Identifies a server in human-readable form. This read-only value is not customizable. |
| Object ID | 1.3.6.1.4.1.2.6.120.1.1.1.1.1.1 |
| Type | OCTET_STRING |
| Default value | An appropriate value for your server installation and platform. |

Managing Your Web Server

EntityObjectID

| | |
|---------------|--|
| Description | Identifies a particular server in machine-readable form, providing a globally unique name among other applications and versions. This read-only value is not customizable. |
| Object ID | 1.3.6.1.4.1.2.6.120.1.1.1.1.1.2.1 |
| Type | OCTET_STRING |
| Default value | 1.3.6.1.4.1.2.6.120.10.1 |

EntityContact

| | |
|---------------|---|
| Description | Indicates who to contact if a problem or question about this running server arises. It is human-readable and frequently contains the e-mail address of the on-site system administrator responsible for server maintenance. The value for EntityContact may be customized with the WebMasterEmail directive in the httpd.conf file. |
| Object ID | 1.3.6.1.4.1.2.6.120.1.1.1.1.1.3.1 |
| Type | OCTET_STRING |
| Default value | webmaster |

EntityProtocol

| | |
|---------------|---|
| Description | Identifies the exact protocol and its version that a particular server supports. For a Web server, the protocol is HTTP. This read-only identifier is in machine-readable form and is not customizable. |
| Object ID | 1.3.6.1.4.1.2.6.120.1.1.1.1.1.4.1 |
| Type | OCTET_STRING |
| Default value | 1.3.6.1.4.1.2.12.1 |

EntityProtocolVersion

| | |
|---------------|--|
| Description | This human-readable string identifies the protocol this server supports and the protocol version. Similar to EntityProtocol. This read-only value is not customizable. |
| Object ID | 1.3.6.1.4.1.2.6.120.1.1.1.1.1.5.1 |
| Type | OCTET_STRING |
| Default value | HTTP/1.1 |

EntityName

| | |
|-------------|---|
| Description | This human-readable string provides the name of the host this Web server runs on. The value for EntityName may be customized with the HostName directive in the httpd.conf file. It is read-only and set by system-specific code at server initialization time. |
|-------------|---|

Managing Your Web Server

Object ID 1.3.6.1.4.1.2.6.120.1.1.1.1.1.6.1
Type OCTET_STRING
Default value www.raleigh.ibm.com

EntityAddress

Description This human-readable string provides the IP address of the host this Web server runs on. It is read-only and set by system-specific code at server initialization time.

Object ID 1.3.6.1.4.1.2.6.120.1.1.1.1.1.7.1
Type IpAddress
Default value 0.0.0.0

EntityPort

Description This human-readable string provides the port number this Web server listens to. It is read-only and set by system-specific code at server initialization time.

Object ID 1.3.6.1.4.1.2.6.120.1.1.1.1.1.8.1
Type INTEGER
Default value 0

EntityType

Description This machine-readable integer differentiates between several server roles.

Possible values are:

- 1 Simple or normal HTTP server
- 2 Proxy server
- 3 Caching server

It is read-only and set by system-specific code at server initialization time. The information is taken from the httpd configuration file.

Object ID 1.3.6.1.4.1.2.6.120.1.1.1.1.1.9.1
Type INTEGER
Default value 1

CurrentThreads

Description Indicates how many threads the server has currently. The total number of active threads is the sum of the MIB values, applInboundAssociations and applOutboundAssociations. This information is read-only.

Object ID 1.3.6.1.4.1.2.6.120.1.1.1.1.1.10.1
Type INTEGER

Managing Your Web Server

Default value 0

MaxThreads

Description Indicates the maximum number of threads the server can have. To affect server performance, use the MaxActiveThreads directive to modify the value for MaxThreads in the httpd.conf file. This is read-only information.

Object ID 1.3.6.1.4.1.2.6.120.1.1.1.1.11.1

Type INTEGER

Default value 0

MinThreads

Description Indicates the minimum number of threads the server can have. This is read only information.

Object ID 1.3.6.1.4.1.2.6.120.1.1.1.1.12.1

Type INTEGER

Default value 1

SummaryRequests

Description Indicates the total number of request the server received plus the total number of requests the server generated (for example, as a proxy server). This read-only information is updated as the server runs.

Object ID 1.3.6.1.4.1.2.6.120.1.1.2.1.1.1.1

Type INTEGER

Default value 0

SummaryRequestErrors

Description Indicates the total number of request errors detected by the server. This read-only information is updated as the server runs.

Object ID 1.3.6.1.4.1.2.6.120.1.1.2.1.1.2.1

Type INTEGER

Default value 0

SummaryRequestDiscards

Description Indicates the total number of requests discarded by the server (for any reason). This read-only information is updated as the server runs.

Object ID 1.3.6.1.4.1.2.6.120.1.1.2.1.1.3.1

Type INTEGER

Default value 0

Managing Your Web Server

SummaryResponses

| | |
|---------------|---|
| Description | Indicates the total number of responses generated or received by this server. This read-only information is updated as the server runs. |
| Object ID | 1.3.6.1.4.1.2.6.120.1.1.2.1.1.4.1 |
| Type | INTEGER |
| Default value | 0 |

SummaryResponseDiscards

| | |
|---------------|--|
| Description | Indicates the total number of responses discarded by the server. This read-only information is updated as the server runs. |
| Object ID | 1.3.6.1.4.1.2.6.120.1.1.2.1.1.5.1 |
| Type | INTEGER |
| Default value | 0 |

SummaryInUnknowns

| | |
|---------------|---|
| Description | Indicates the total number of unknown messages received by this server. This read-only information is updated as the server runs. |
| Object ID | 1.3.6.1.4.1.2.6.120.1.1.2.1.1.6.1 |
| Type | INTEGER |
| Default value | 0 |

SummaryInBytes

| | |
|---------------|--|
| Description | Indicates the total number of bytes received by this server. This read-only information is updated as the server runs. |
| Object ID | 1.3.6.1.4.1.2.6.120.1.1.2.1.1.7.1 |
| Type | INTEGER |
| Default value | 0 |

SummaryOutBytes

| | |
|---------------|--|
| Description | Indicates the total number of bytes sent out by this server. This read-only information is updated as the server runs. |
| Object ID | 1.3.6.1.4.1.2.6.120.1.1.2.1.1.8.1 |
| Type | INTEGER |
| Default value | 0 |

TotalTimeouts

| | |
|-------------|---|
| Description | Indicates the total number of timeouts on the Web server. This read-only information is updated as the server runs. |
| Object ID | 1.3.6.1.4.1.2.6.120.1.1.2.4.0 |

Managing Your Web Server

Type INTEGER

Default value 0

LastTimeoutEntityIndex

Description This value is for future extensibility and provides support for the Application Table. This read-only value is always 1 and is not customizable.

Object ID 1.3.6.1.4.1.2.6.120.1.1.2.5.0

Type INTEGER

Default value 1

LastTimeoutRemoteAddress

Description Provides the IP address of the machine that timed out last. This read-only value is updated by server code as the server runs.

Object ID 1.3.6.1.4.1.2.6.120.1.1.2.6.0

Type IpAddress

Default value 0.0.0.0

appName

Description The name that the network service application is known by. This read-only value is not customizable.

Object ID 1.3.6.1.2.1.27.1.1.2.1

Type OCTET_STRING

Default value Internet Connection Server *platform*

appDirectoryName

Description The X.500 name for Web server. This read-only value is not customizable and is currently not supported by Internet Connection Secure Server Web server.

Object ID 1.3.6.1.2.1.27.1.1.3.1

Type OCTET_STRING

Default value Not available

appVersion

Description The version of software the server is running. This human-readable value is read-only and not customizable.

Object ID 1.3.6.1.2.1.27.1.1.4.1

Type OCTET_STRING

Default value 4.2

appUptime

Managing Your Web Server

Description This value is how long the server has been up. This is a read-only value.

Object ID 1.3.6.1.2.1.27.1.1.5.1

Type TimeTicks

Default value 0

applOperStatus

Description Indicates the operational status of the Web server. The Internet Connection Secure Server sets this value to *up* at server startup. It is currently a read-only value.

Additional standardized values for this MIB variable include down, halted, congested, and restarting. These values may be used in the future.

Standardized values include:

- 1 Up - indicates that the server is operational and available.
- 2 Down - indicates that the Web server is not available.
- 3 Halted - indicates that the Web server is operational but not available.
- 4 Congested - indicates that the server is operational but no additional inbound associations can be accommodated.
- 5 Restarting - indicates that the server is currently unavailable but is in the process of restarting and will be available soon.

Object ID 1.3.6.1.2.1.27.1.1.6.1

Type INTEGER

Default value 1

applLastChange

Description Indicates how long from when the server came up (applUptime) that the applOperStatus changed. Currently this will always be 0 because applOperStatus is only set to *up* at server startup. This is a read-only value.

Object ID 1.3.6.1.2.1.27.1.1.7.1

Type TimeTicks

Default value 0

applInboundAssociations

Description Indicates the number of inbound connections currently running or how many threads are processing received requests. This is a read-only value.

Object ID 1.3.6.1.2.1.27.1.1.8.1

Managing Your Web Server

Type Gauge32

Default value 0

applOutboundAssociations

Description Indicates the number of outbound connections that the server is currently handling or how many threads are processing outbound requests. This value is 0 if the server is not acting as a proxy server. This is a read-only value.

Object ID 1.3.6.1.2.1.27.1.1.9.1

Type Gauge32

Default value 0

applAccumulatedInboundAssociations

Description Indicates the total number of server's inbound connections until this time. This is a read-only value.

Object ID 1.3.6.1.2.1.27.1.1.10.1

Type Gauge32

Default value 0

applAccumulatedOutboundAssociations

Description Indicates the total number of server's outbound connections until this time. This value is 0 if the server is not acting as a proxy server. This is a read-only value.

Object ID 1.3.6.1.2.1.27.1.1.11.1

Type Counter32

Default value 0

applLastInboundActivity

Description Indicates the time since applUptime that the last inbound connection was made. This is a read-only value.

Object ID 1.3.6.1.2.1.27.1.1.12.1

Type TimeTicks

Default value 0

applLastOutboundActivity

Description Indicates the time since applUptime that the last outbound connection was made. This is a read-only value.

Object ID 1.3.6.1.2.1.27.1.1.13.1

Type TimeTicks

Default value 0

applRejectedInboundAssociations

Managing Your Web Server

Description Indicates the total number of requests the server has rejected. This is a read-only value.

Object ID 1.3.6.1.2.1.27.1.1.14.1

Type Counter32

Default value 0

applFailedOutboundAssociations

Description Indicates the total number of the server's outbound requests that failed. This is a read-only value.

Object ID 1.3.6.1.2.1.27.1.1.15.1

Type Counter32

Default value 0

Note: The timestamp values for the Internet Connection Secure Server MIB variables, applLastChange, applLastInboundActivity, and applLastOutboundActivity, vary from RFC 1565. In RFC 1565, timestamps are relative to sysUpTime. These three Internet Connection Secure Server timestamp values are relative to applUptime.

Creating an e-mail address to receive SNMP problem reports

The Internet Connection Secure Server provides a default e-mail address, webmaster, for the Web server administrator to receive problem reports from SNMP. Use the WebMasterEmail directive to customize the mail address. The typical format for this value is user@rootname. For more information about the WebMasterEmail directive, see "WebMasterEmail - Creating an e-mail address to receive SNMP problem reports" on page 151.

Enabling SNMP on your OS/2 system

To get a DPI-capable SNMP agent for OS/2, install the SystemView Agent Developers toolkit for OS/2, which can be downloaded from the <http://www.software.ibm.com/download/> URL.

To configure the SystemView SNMP agent, follow the instructions in "Providing a security password for SNMP" on page 233.

Providing a security password for SNMP

You can create community names (passwords). The default SNMP community name is `public`.

Before creating passwords for server communities, configure the SystemView SNMP agent.

1. From your desktop, click on the **SystemView Agent** icon.
2. Click on the **OS/2 Agent Configuration** icon.

The SNMP Configuration Notebook displays.

3. On the front page of the SNMP Configuration Notebook, enter the following values before testing SNMP access.
 - Community name: `public`
 - Hostname: `0.0.0.0`
 - Netmask: `0.0.0.0`

These values allow any host in any network to access the SNMP MIB variables. After you have verified that these values work, you can change them according to your requirements.

4. Under the SNMP tab, start `snmpd` with the **-dpi** and **-transport udp** options selected. Other options can be specified as well.

For more information about configuring the SystemView SNMP agent, see the on-line *SystemView Agent for OS/2 User's Guide*.

Use the `SNMPCommunityName` directive to define the password used between the Internet Connection Secure Server DPI subagent and the SNMP agent. The default is `public`. If you change this value, you must also add the new community name to the SystemView Agent SNMP Configuration Notebook. For more information about the `SNMPCommunityName` directive, see "SNMPCommunityName - Providing a security password for SNMP" on page 150.

Enabling and disabling SNMP support

Use the `SNMP` directive to enable or disable SNMP support. To enable SNMP support, change the `SNMP` value to `on`. The default `SNMP` value is `off`.

For more information about the `SNMP` directive, see "Turning the SNMP support on and off from the `httpd` command" on page 234.

Managing Your Web Server

Turning the SNMP support on and off from the httpd command

Use these flags to turn the SNMP support in the Internet Connection Secure Server on and off.

The `-snmp` flag turns the SNMP support on. The `-nosnmp` flag turns the SNMP support off.

This overrides what is defined in the `httpd.conf` file.

For more information about the `httpd` command, see “`httpd` command” on page 298.

Monitoring server performance and status

The Internet Connection Server Activity Monitor allows you to display server and network performance and status statistics, and access log entries, without being on the same machine that is running the server. This option provides significantly more information than opening the console window.

Using the Server Activity Monitor function

You do not need to enable or configure the Server Activity Monitor function. By default, the Server Activity Monitor is enabled by the following Service directive in the configuration file:

```
Service /Usage* INTERNAL:UsageFn
```

To view server statistics, network statistics, and access log entries, use either of the following methods:

- Use any browser and specify URL **`http://your.server.name/Usage/Initial`**, where *your.server.name* is the fully qualified name of your host, for example, **`www.ibm.com`**.
- Click **Configuration and Administration Forms** on the server Front page. Then click **Server Activity Monitor** in the **System Management** section.

The following sections describe the type of information that is available and provide hints and tips on monitoring certain statistics. To update the statistics on a page, click **Refresh**.

Server activity statistics

Figure 9 shows an example of the server statistics that are displayed on the Basic Status page.

| Thread counts | | Request statistics | | Throughput statistics | | Connection counts | |
|--------------------------|----|-------------------------|-----|-------------------------------------|---------------|------------------------------|---|
| Threads running: | 40 | Requests processed: | 142 | Response time for local files: | 3 seconds | Active inbound connections: | 1 |
| Threads idle: | 39 | Request errors: | 3 | Response time for proxied requests: | Not available | Active outbound connections: | 0 |
| Minimum allowed threads: | 1 | Requests discarded: | 8 | Bytes received: | 62K | | |
| Maximum allowed threads: | 40 | Requests proxied today: | 0 | Bytes sent: | 568K | | |
| | | Proxy cache hit rate: | 0% | Unknown Bytes Received: | 0K | | |
| | | Responses processed: | 145 | | | | |
| | | Responses discarded: | 0 | | | | |

Figure 9. Example of Basic Status page statistics

Hints and Tips:

- If the number of idle threads is low, you may need to increase the number of threads that are available to the server.
- To monitor server response, use the **Response time for local files** statistic.
- To monitor traffic, use the **Requests processed**, **Bytes received**, and **Bytes sent** statistics.

Network activity statistics

Figure 10 on page 236 shows an example of the network statistics that are displayed on the Network Status page.

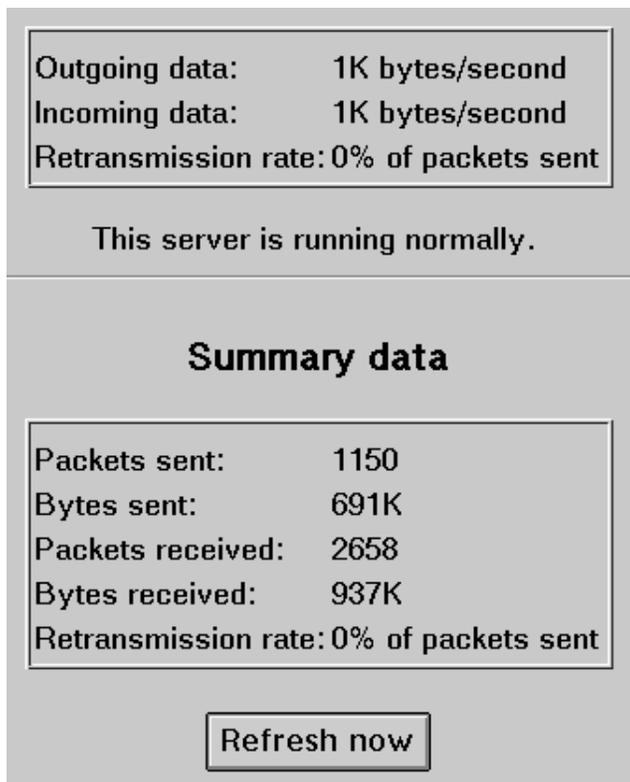


Figure 10. Example of Network Status page statistics

Note: Incoming and outgoing data values include data received and sent by the server, by other applications on the machine, and overhead added by the subsystem.

Access log

The access log page displays the 20 most recent entries in the access log. For more information on the access log, go to “Tailoring the logs your server keeps” on page 153.

Part 3. Security

Chapter 9. Making your communications secure

If you have a secure server, you can use the security built into it to provide a secure environment for you and your users to conduct business transactions.

This chapter provides an overview of security and explains the different security protocols.

| | |
|---|-----|
| Security concepts | 239 |
| The Internet is a postal service | 240 |
| What is security? | 240 |
| What is encryption? | 241 |
| What is authentication? | 242 |
| Using the security built into the server | 244 |
| Managing your keys, certificates, and trusted roots | 245 |
| Working with keys | 245 |
| Setting the default key | 248 |
| Deleting a key | 249 |
| Exporting a key | 249 |
| Importing a key | 250 |
| Obtaining a certificate | 251 |
| Using trusted root keys | 252 |
| Designating a key as a trusted root | 253 |
| Removing a trusted root key | 253 |

Security concepts

The Internet was not originally designed to protect sensitive or confidential data. However, with the advent of electronic commerce, such as online shopping and online bill paying, information privacy becomes necessary. In addition, intra-company communications over the Internet often contain confidential product information that needs to be protected from general access. Routing computers can listen to and record any communications that pass through.

Security

The Internet is a postal service

Think of the Internet as a new postal service. This new service operates at blinding speed, sorting and routing messages through various post offices in seconds.

Unfortunately, this new postal service has an awkward twist to it. Anyone and everyone can be a postmaster, and there is nothing to prevent a postmaster from reading, copying, and even altering the mail as it passes through the post office. Another drawback is the lack of receiver accountability, for items such as registered letters that require signatures or return receipts. As a result, you cannot be sure that the intended recipient received the mail you sent.

Because the Internet was designed to be an open system, it allows any computer on the network to see the messages passing through. You can safely estimate that 20% of the messages you send on the Internet will be copied and stored somewhere by someone else for future reference. Because most messages are sent as plain block text, there is no limit to the different software platforms that can read, copy, and possibly alter the messages you send.

What is security?

You would never think of sending your medical records or paying your bills by postcard. By the same token, few people want to use the Internet, in its present form, for commercial exchanges without additional security.

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

- Confidentiality
- Integrity
- Accountability
- Authenticity

Confidentiality

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

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

Security

integrity, you have no guarantee that the message you sent matches the message received. Encryption and **digital signature** ensure integrity.

Accountability

Accountability means that both the sender and the receiver agree that the exchange took place. Without accountability, the addressee can easily say the message never arrived. Digital signature also ensures accountability.

Authenticity

Authenticity means that you know who you are talking to and that you can trust that person. Without authenticity, you have no way to be sure that anyone is who they say they are. **Authentication** ensures authenticity.

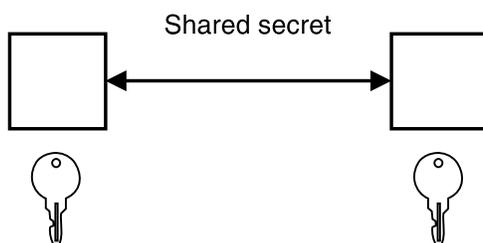
What is encryption?

Encryption in its simplest form is scrambling a message so that it cannot be read until it is unscrambled later by the receiver. The sender uses an algorithmic pattern, or **key**, to scramble, or encrypt, the message. The receiver has the decryption key. Encryption ensures confidentiality in transmissions sent over the Internet.

There are two kinds of keys that can be used for encryption (as well as for digital signature and authentication):

- Symmetric
- Asymmetric

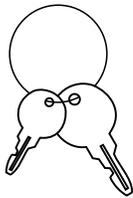
Symmetric keys follow an age-old model of the sender and receiver sharing some kind of pattern. This same pattern is then used by the sender to encrypt the message and by the receiver to decrypt the message. You may have used this model when you decoded the secret message on the back of a cereal box using your secret decoder ring.



Security

The risk involved with symmetric keys is that you have to find a safe transportation method to use when sharing your secret key with the people with which you want to communicate.

With **asymmetric** keys, you create a **key pair**.



The key pair is made up of a **public key** and a **private key**, which are different from each other. The private key holds more of the secret encryption pattern than the public key.

As a sender, you can then broadcast your public key to whomever you want to communicate securely. You hold on to the private key and protect it with a password.

Unlike symmetric keys, the private key and the public key are not the same. As a result, only you can decrypt a message that has been encrypted with your public key, because only you have the private key.



A protocol like Secure Sockets Layer (SSL) uses both public key cryptography and symmetric key cryptography. Public key cryptography is used for the TCP/IP handshake. During the handshake the master key is passed from the client to the server. The client and server make their own session keys using the master key. The session keys are used to encrypt and decrypt data for the remainder of the session.

What is authentication?

Authentication is the process used to verify identity, so that you can make sure that others are who they say they are. There are two ways in which the server uses authentication:

- Digital signature

Security

- Digital certificates

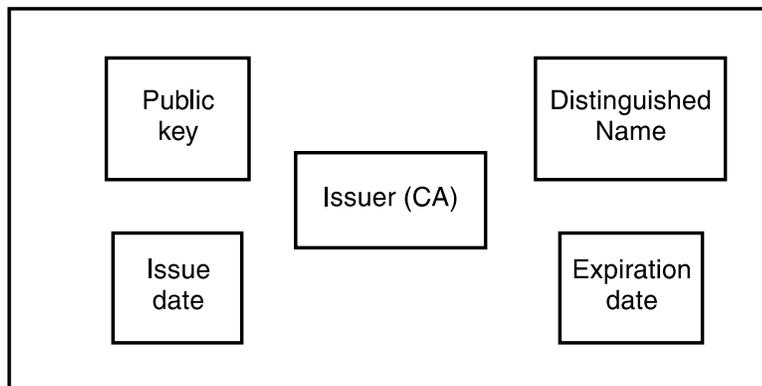
A digital signature is a unique mathematically computed signature that ensures accountability. You can think of a digital signature as being like a credit card with your picture on it. But how do you know if the person sending you a message is who he says he is?

You look at the sender's **digital certificate**. A digital certificate is like a credit card with a picture of the bank president with his arm around you. A merchant will trust you more because not only do you look like the picture on the credit card, the bank president trusts you, too.

You base your trust for the authenticity of the sender on whether you trust the third party (a person or agency) that certified the sender. The third party or **certification authority (CA)** issues digital certificates.

A certificate is made up of:

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



The Distinguished Name is the name and address of a person or organization. At a minimum, it is someone's name. You enter your Distinguished Name as part of requesting a certificate. The digitally-signed certificate includes not only your own Distinguished Name, but the Distinguished Name of the CA.

CAs broadcast their public key and Distinguished Name bundled together so that people will add them as a **trusted root** key to their Web servers and browsers. When you designate the public key and certificate from a certification authority to be a trusted root key means that your server will trust anyone who has a certificate from that certification authority. You may have many trusted roots as part of your server. In fact,

Security

the server includes several default trusted root keys, and you can add others as needed.

In order to communicate securely, the receiver in a transmission must trust the CA that issued the certificate that the sender is using. This is true whether the receiver is a Web server or browser. As a result, any time a sender signs a message, the receiver must have the corresponding CA certificate and public key designated as a trusted root key.

Using the security built into the server

SSL is a security protocol that was developed by Netscape Communications Corporation, along with RSA** Data Security, Inc. This protocol ensures that data transferred between a client and a server remains private. It allows the client to authenticate the identity of the server. In addition, SSL V3 allows the server to authenticate a client. If SSL client authentication is set up, the server requests the client's certificate for any **https** request. The server establishes a secure session whether or not the client has a valid certificate. SSL client authentication parameters can be specified in protection setups, ACL files, or both as described in Chapter 7, "Protecting your server" on page 201. If you specify SSL client authentication parameters, the server validates the DN information in the parameters with the DN information in the client's certificate before serving the document.

Once your server has a digital certificate, SSL-enabled browsers like the IBM Secure WebExplorer and Netscape** Navigator can communicate securely with your server using SSL. With SSL, you can easily establish a security-enabled Web site on the Internet or on your corporate TCP/IP network.

SSL uses a security handshake to initiate the TCP/IP connection between the client and the server. During the handshake, the client and server agree on the security keys that they will use for the session and the algorithms they will use for encryption and to compute message digests or hashes. The client authenticates the server. In addition, if the client requests a document protected by SSL client authentication, the server requests the client's certificate. After the handshakes, SSL is used to encrypt and decrypt all of the information in both the **https** request and the server response, including:

- The URL the client is requesting
- The contents of any form being submitted
- Access authorization information like user names and passwords
- All data sent between the client and the server

For more information on how to use SSL, see Chapter 10, "Using Secure Sockets Layer (SSL)" on page 255.

Managing your keys, certificates, and trusted roots

The server uses the public key cryptography from RSA Data Security, Inc., for encryption, message digest, and authentication. RSA public key cryptography is commonly used in the computer industry.

We provide forms that you can use to manage your keys, **key ring** file passwords, certificates, and trusted roots. A key ring file is a file where you keep public keys, private keys, certificates, and trusted root keys.

Working with keys

As discussed earlier in this chapter, public key cryptography uses a pair of asymmetric keys for encryption and decryption.

When you create a key pair, you are asked to provide the name of the key ring where you want to keep the key pair and a password for the key ring. Then, whenever you want to use a Key Management form, such as Designate Trusted Root Keys, you'll be asked for the key ring password.

The default trusted root keys that are part of the server will automatically be added any time you create a key ring.

The server also provides you with the ability to:

- Change your key ring password
- Set a private key as the default
- Delete keys
- Export a key by copying it to a file
- Import a key from an exported copy and add it to a key ring

You can access these key management functions from the Key Management form. To access this form:

- 1** Go to the front page of your server and click **Configuration and Administration forms**.
- 2** You are prompted for your user name and password. Type them in and click **OK**.
- 3** The Configuration and Administration forms page appears. Page down until you find the topic Security. Select **Key Management**.

Security

Key Management

Use the key management forms to manage your keys and certificates. This form shows the current key ring that you'll be working with.

Current key ring: D:\WWW\BIN\keyfile.kyr

Specify the key ring password.

Key Ring Password

Choose the key management task you want to perform for the current key ring.

- Change Password - Change key ring password
- Manage Keys - Delete keys or make a key the default in this key ring
- Export Keys - Transfer key pair or certificate to another key ring or computer
- Import Keys - Transfer key pair or certificate to this key ring
- Request Certificate - Request certificate for an existing key
- Designate Trusted Root Keys - Designate keys as trusted root keys
- Remove Trusted Root Keys - Remove trusted root key designation

The key management forms require that you enter the key ring password before you can use them. If you have multiple key rings, always make sure that you are making changes to the correct key ring by checking the key file name and path displayed on the form. When you complete your key management tasks, always make sure that the key ring that contains the server's keys you want to use for secure communications is the current key ring. You can use the Security Configuration form to designate a key ring as the current key ring.

When you apply the changes you've made to a security form, you will receive a confirmation message that the changes have been successfully completed. You can make changes to other Configuration and Administration forms. However, when you are ready for the changes you've made to any security forms to take effect, you must stop the server and then start it again. The server will not pick up your changes if you only restart it.

Changing your key ring password

Use the Change Password form to change the password for the current key ring file.

Change Password

Use this form to change the password for the current key ring.

Current key ring: D:\WWW\BIN\keyfile.kyr

Specify the old password. Then, specify the new password. If you check **Automatic login**, the password is automatically specified when the server is started. For non-interactive startup, make sure this box is checked. For your CA key ring, make sure this box is not checked. If your CA keys are compromised, all the certificates you have issued are also compromised.

Old password

Password

Password (for verification)

Automatic login

Choose whether or not you want the password to expire. If you choose **Password expires in**, specify the number of days.

Password does not expire

Password expires in days

The current key ring file name is displayed. To change your password:

Old password: Enter the current password used to protect the key ring file.

Specify a new password: Use this part of the form to specify the new password to protect your key ring. The password must be from the U.S. English character set.

The password is important because it protects a private key. The server uses its private key to decrypt messages from clients and to sign messages to clients. The CA private key is used to sign the client and server certificates processed by the certutil command.

The security of a private key depends upon this password. Here are some commonly accepted rules for specifying passwords:

- Make your password at least six characters.
- Make sure your password doesn't spell a word.
- Make sure your password doesn't consist of publicly obtainable information about you; for example, the initials and birth date for you, your spouse, or children.
- Include at least two, nonconsecutive numbers in your password.

Type the password twice to ensure that you have typed it correctly.

If you must record the password, make sure it is stored in a well-secured place.

The key ring password must be specified when the server is started. Check **Automatic login** if you want the server to stash the password and specify it for you whenever the

Security

server is started. To stash the password, the server encrypts the key ring password and puts it in a file that has the same name as the key ring file, except the extension is .sth.

Stashing the server's key ring password is less secure so it's generally a good practice not to stash it. However, there may be cases where you need to stash the password; for example, if you have a remote server and you want it to automatically restart after a power failure.

If you are changing the CA key ring password, do not stash it.

Expiration: Select whether you want the password to never expire or the length of time you want the password to be active. If you do select an expiration date, a message is written to the error log when the password expires.

Setting the default key

Use the Manage Keys form to set a private key as the default key. The default key should be the private key the server uses for its secure communications.

Manage Keys

Use this form to delete a key or to make a key the default key for the current key ring.

Current key ring: D:\WWW\BIN\keyfile.kyr
Current default key: mykey

Choose the key you want to work with. You cannot make a trusted root key (designated with an "*") the default key for the key ring. The default key should be the key the server uses for its secure communications.

Keys

| |
|---|
| mykey |
| * RSA Secure Server Certificate Authority |
| * Netscape Test Certificate Authority |
| * RSA Low Assurance Certificate Authority |
| * Verisign Persona Certificate Authority |

Choose the action you want to take.

Set as default
 Delete

The current key ring file name and default key are displayed. To set a key as the default:

Specify key: Select the key you want to set as default, as long as it is not a trusted root key. You cannot set a trusted root key as a default because trusted root keys cannot send and receive messages. The trusted roots are marked with an asterisk (*).

Set as default: Select this radio button to set the key as the default.

Deleting a key

Use the Manage Keys form to delete a key from the current key ring file.

Manage Keys

Use this form to delete a key or to make a key the default key for the current key ring.

Current key ring: D:\WWW\BIN\keyfile.kyr
 Current default key: mykey

Choose the key you want to work with. You cannot make a trusted root key (designated with an "™") the default key for the key ring. The default key should be the key the server uses for its secure communications.

Keys

mykey
 * RSA Secure Server Certificate Authority
 * Netscape Test Certificate Authority
 * RSA Low Assurance Certificate Authority
 * Verisign Persona Certificate Authority

Choose the action you want to take.

Set as default
 Delete

The current key ring file name and default key are displayed. To delete a key:

Specify key: Select the key you want to delete. The default trusted roots are marked with an asterisk (*).

Delete: Select this radio button to delete the key.

Exporting a key

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.

Use the Export Keys form to export a key pair or certificate to a file.

Security

Export Keys

Use this form to export a key pair or certificate from the current key ring to another key ring or computer.

Current key ring: D:\WWW\BIN\keyfile.kyr

Choose the key pair or certificate that you want to export. Trusted root keys are designated with an "*".

Keys and Certificates

| |
|---|
| mykey |
| * RSA Secure Server Certificate Authority |
| * Netscape Test Certificate Authority |
| * RSA Low Assurance Certificate Authority |
| * Verisign Persona Certificate Authority |

Specify the fully qualified path and file name of the file to which you want to export the key pair or certificate. Also specify the password that will be used to encrypt the file.

Export to file

Armor password

Armor password (for verification)

To export:

Specify key: Select the key pair or certificate that you want to export. Trusted root keys are designated with an "*" You may need to transfer a key pair for your own use; however, keep in mind that a key pair contains the private key, which should never be transmitted to others. The private key is what the server uses to sign messages and also to decrypt messages that clients have encrypted with the server's public key. If you are acting as your own CA for a private Web network, the CA private key is used to sign the certificates you process as CA. If the CA private key is compromised, then all the certificates issued by you as a CA are also compromised.

Export to file: Enter the name of the file to which you want to export the key pair or certificate.

Armor password: Enter the password you want use to protect the export file. Then enter it again for verification. The export file will be encrypted using this password.

Importing a key

Use the Import Keys form to import a key pair or certificate from a file and add it to a key ring file.

Import Keys

Use this form to import a key pair or certificate into the current key ring.

Current key ring: D:\WWW\BIN\keyfile.kyr

Specify the fully qualified path and file name of the file from which you want to import the key pair or certificate. Also specify the password that was used to encrypt the exported file.

Name of file to import

Armor password

To import a key pair or certificate:

Name of file to import: Enter the name of the file from which you want to import the key pair or certificate.

Armor password: Enter the password that was used to protect the export file.

Obtaining a certificate

Your public key must be tied to a digitally signed certificate from a (CA). There are two different ways that you can obtain a certificate:

- Buy a certificate from a CA
- Issue yourself a certificate using the certutil command

It's not a good practice to share certificates among servers. As you'll see later, the server's keys are kept on a key ring with the server's certificate. You do not want servers to share a private key, particularly if they are running on different machines. A private key should never be communicated to others.

VeriSign** certificates can't be shared among servers on different machines.

Buying a certificate from a CA

You can buy a signed certificate by submitting a certificate request to a CA. VeriSign, Inc. will issue certificates to Internet Connection Secure Server customers. We expect that there may be other certification authorities over time.

VeriSign will expect you to prove who you are before they will issue you a certificate. Although the approval process is necessary to protect you, your organization, and VeriSign, it may take longer than you would like. VeriSign will digitally sign your certificate request and return the unique certificate to you through e-mail. After you have received the certificate into the file system, you can use the Configuration and Administration forms to receive the certificate into a key ring.

Security

For more detailed step-by-step information on how to request a certificate, see Chapter 10, “Using Secure Sockets Layer (SSL)” on page 255.

Issuing certificates yourself

If you act as a CA, you can sign your own or anyone else's certificate request. This is a good choice if you only need the certificates within your private Web network and not for outside Internet commerce. Clients must have browsers, such as Secure WebExplorer and Netscape Navigator, that can receive your CA certificate and designate it as a trusted root.

For example, if your company is working on a top secret project and you want to allow access only to a carefully controlled group of key people, you might want to issue them certificates so that they can share information covertly and safely.

If you choose to issue your own certificates, you will need to give the file that contains the signed certificate to users. They can then use read the certificate and designate it as an additional trusted root.

For more detailed step-by-step information on how to issue certificates, see Chapter 11, “Acting as a certification authority for a private Web network” on page 269.

Using trusted root keys

In order for the server to trust a request coming from a client, it must trust the CA that issued the client certificate. If you want to trust certificates coming from a CA other than the default trusted roots included with the server, you must receive a certificate from that CA and designate it as a trusted root.

With the remote configuration forms, you can:

- Designate a public key and certificate as a trusted root key
- Remove trusted root status from a public key and certificate

You can access these trusted root functions from the Key Management form. To access this form:

- 1** Go to the front page of your server and click **Configuration and Administration forms**.
- 2** You are prompted for your user name and password. Type them in and click **OK**.
- 3** The Configuration and Administration forms page appears. Page down until you find the topic Security. Select the Key Management form.

Designating a key as a trusted root

Use the Designate Trusted Root Keys form to designate a trusted root key.

Designate Trusted Root Keys

Use this form to designate a key in the current key ring as a trusted root key.

Current key ring: D:\WWW\BIN\keyfile.kyr

Choose the key you want to designate as a trusted root. Only the public key of a certification authority should be designated as a trusted root.

Keys

mykey

The current key ring file is displayed. To designate a trusted root key:

Specify key: Select the key you want to designate as a trusted root. Do not designate keys as trusted root keys if you want the server to use them for secure communications. Trusted root keys cannot send or receive messages.

Removing a trusted root key

You may choose not to accept client requests from clients that have certificates issued by a specific CA.

Use the Remove Trusted Root Keys form to remove trusted root status from a key.

Note: This does not delete the key from your key ring file. See “Deleting a key” on page 249 to delete a key.

Remove Trusted Root Keys

Use this form to remove trusted root status from a key in the current key ring.

Current key ring: D:\WWW\BIN\keyfile.kyr

Choose the key you want to remove as a trusted root. **Note:** This does not delete the key from the key ring. You use the Manage Keys form to delete keys.

Keys

- * RSA Secure Server Certificate Authority
- * Netscape Test Certificate Authority
- * RSA Low Assurance Certificate Authority
- * Verisign Persona Certificate Authority

Security

The current key ring file is displayed. To remove trusted root status from a key:

Specify key: Select the key you want to remove as a trusted root.

Chapter 10. Using Secure Sockets Layer (SSL)

This chapter explains how to use SSL to make the server's communications secure.

For an overview of security and an explanation of SSL, see Chapter 9, "Making your communications secure" on page 239.

| | |
|---|-----|
| Overview of SSL security | 255 |
| Step 1. Setting up SSL | 256 |
| Fill in Security Configuration form | 257 |
| Fill in Create Key and Request Certificate form | 258 |
| Fill in Receive Certificate Form | 265 |
| Step 2. Specifying SSL client authentication in protection setups and ACL files | 267 |
| Step 3. Using SSL with your server | 267 |

Overview of SSL security

HTTPS is a unique protocol that combines SSL and HTTP. You need to specify **https://** as an anchor in HTML documents that link to SSL-protected documents. A client user can also open a URL by specifying **https://** to request an SSL-protected documents.

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

A browser that does not have support for HTTP over SSL will naturally not be able to request URLs using HTTPS. The non-SSL browsers will not allow submission of forms that need to be submitted securely.

Step 1. Setting up SSL

To implement security you need to specify the port you'll use for secure network communications. If the server needs to ensure who a client is before responding, you need to set up SSL client authentication. When you set up SSL client authentication, the server requests a client's certificate whenever the client makes an **https** request.

Also, the server must have a public-private key pair and a certificate.

The server uses its private key to sign messages to clients. The server sends its public key to clients so that they can encrypt messages to the server, which the server decrypts with its private key.

To send its public key to clients, the server needs a certificate. The certificate contains the server's public key, the Distinguished Name associated with the server's certificate, the serial number of the certificate, and the expiration date of the certificate.

A certificate is issued by a certification authority (CA), who verifies that you are who you say you are.

To conduct commercial business on the Internet, you would use a CA, such as VeriSign, Inc., who is widely known by clients and servers to get a server certificate. If you plan to implement SSL client authentication, clients can also get their certificates from a CA such as Verisign. The VeriSign home page at <https://www.verisign.com/enroll.s> contains instructions for how clients can obtain certificates.

For a private Web network within your own company, university, or group, you could, with your Internet Connection Secure Server, be your own CA. To learn how to become your own CA and process certificates for this server and other clients and servers, see Chapter 11, "Acting as a certification authority for a private Web network" on page 269.

Use the following **Configuration and Administration forms** to set up SSL security:

- Security Configuration form to specify the port that will be used for SSL
- Create Key and Request Certificate form to create the server's public-private key pair and certificate request
- Receive Certificate form to receive the server's certificate into a key ring so that the server can use it for secure network communications

To access the Configuration and Administration forms:

- 1 Go to the front page of your server and click **Configuration and Administration forms**.
- 2 You are prompted for your user name and password. Type them in and click **OK**.

- 3 The **Configuration and Administration forms** page appears. Page down until you find the topic Security. To access any security form listed, click its name.

Fill in Security Configuration form

Use the **Security Configuration** form to specify some of the security information needed for your server.

1. Specify connection options

Use the first part of the **Security Configuration** form to specify your connection for SSL and to set up SSL client authentication.

Security Configuration

Use this form to configure security options for SSL and S-HTTP.

Connection options:

Check the boxes for the kinds of security connections you want. You can have an S-HTTP connection, an SSL connection, or both an S-HTTP connection and an SSL connection.

Important: Be sure to check at least one box. If no box is checked, the server will not start. To recover, you will have to locally edit the configuration file.

Allow HTTP and S-HTTP connections

Allow SSL connections using port

Check the box to allow SSL connections and, optionally, to allow HTTP connections. In addition, if the server needs to verify who a client is before responding, you can check the box to set up SSL client authentication. Authenticating clients increases network traffic. If you set up SSL client authentication, the server requests a certificate when a client makes an **https** request.

2. Process your request

Check your form and make sure you entered each item correctly. Click **Apply** to process the form.

You receive a Confirmation that the form was successfully processed. You can make changes to other Configuration and Administration forms. However, when you are ready for the changes you've made to this form to take effect, you must stop the server and then start it again. The server will not pick up your changes if you only restart it.

For SSL, the server must have a public-private key pair and a certificate.

Using SSL

Fill in Create Key and Request Certificate form

These instructions tell you how to use this form to do one of the following:

- Create a key pair and request a certificate for the server from VeriSign.
- Create a key pair and request a certificate for the server from a CA other than VeriSign. The other CA could be some other outside CA. Or, someone within your organization could act as a CA for a private Web network; for example, you might act as your own CA and the CA for others in your group. If you are acting as a CA, you need to follow the instructions in Chapter 11, “Acting as a certification authority for a private Web network” on page 269. The chapter tells how to become a CA and how to process certificate requests.

To create the server's public-private key pair and request a certificate, use:

- The Create Key and Request Certificate form to choose your CA
- One of the following forms, depending on the CA you choose, to create the server's keys and request its certificate:
 - VeriSign Low Assurance Certificate
 - VeriSign Secure Server Certificate
 - Other Certificate

1. Choose a CA

Use the Create Key and Request Certificate form to specify the CA from whom you want to obtain a certificate.

Create Key and Request Certificate

Choose the certification authority (CA) from whom you want to obtain a certificate. VeriSign is a widely known CA. For information about obtaining a certificate from VeriSign, you can access the [VeriSign home page](#). If you want to use another CA or to act as your own CA for a private Web network, choose Other.

VeriSign (Low Assurance Certificate)
 VeriSign (Secure Server Certificate)
 Other

VeriSign is a widely known CA. Choose one:

- **VeriSign (Low Assurance Certificate):** The level of proof required to get a low assurance certificate is less than for a secure server certificate. You might want to get a low assurance certificate to do a beta test of security with your clients or for some other non-commercial purpose.

Using SSL

- **VeriSign (Secure Server Certificate):** To conduct commercial business on the Internet, you want a secure server certificate. The level of proof required to get a secure server certificate is higher than for a low assurance certificate.
- **Other:** This is any other CA. For example, for a private Web network, your organization could choose to act as its own CA.

2. Process your request

Click **Apply** to process this part of the Create Key and Request Certificate form.

Another part of the Create Key and Request Certificate form appears, depending on which CA you choose:

- VeriSign Low Assurance Certificate
- VeriSign Secure Server Certificate
- Other Certificate

3. Create public-private key pair

Use **Create Key** to create a key pair.

Create Key

Specify a unique, meaningful name, which will be used to identify the public-private key pair. Also specify the size of the key pair and the fully qualified path and file name of the key ring where the key pair will be kept.

| | | | | | |
|----------|---|------|----------------------------------|--|------|
| Key name | <input type="text" value="mykey"/> | Size | <input type="text" value="512"/> | <input type="button" value="↑"/> <input type="button" value="↓"/> | bits |
| Key ring | <input type="text" value="D:/WWW/BIN/keyfile.kyr"/> | | | | |

Specify key name: Specify a meaningful, unique name to identify the key pair. The **key name** is the label that identifies the key pair and certificate in the key ring. You may use non-alphanumeric characters in key names; however, keep in mind that some platforms have special uses for some of these characters.

Specify key ring: Specify the fully qualified path and file name for the key ring file. A **key ring** is a file where the server keeps one or more key pairs and certificates. If you specify a file that doesn't exist, the server creates the file for you.

Specify the size of the key pair: Specify the size of the key pair in bits after considering these factors:

- The more bits you specify for the key pair, the more secure your communications.
- The more bits you specify, the more processing time required for encryption and decryption.

Using SSL

- With export versions of the server, you can create a key pair up to 512 bits. With U.S.-Canadian versions, you can create a larger key pair up to 1024 bits; however, if you want clients outside the United States to encrypt messages to the server, U.S. export rules require that the key pair be no larger than 512 bits. If you want clients outside the United States only to verify your server's signature or sign messages, the key pair can be larger than 512 bits.

4. Specify key ring password

Use this part of the form to specify the password that protects the key ring.

Key Ring Password

Specify a password for the key ring. The key ring password must be specified each time the server is started. If you check **Automatic login**, the password is automatically specified when the server is started. For non-interactive startup, make sure this box is checked.

Password

Password (for verification)

Automatic login

The password must be from the U.S. English character set.

The password is important because it protects the server's private key. The server's private key is the only key that can decrypt messages encrypted with the server's public key. Also, the server signs documents with its private key.

The security of the server's private key depends upon this password. Here are some commonly accepted rules for specifying passwords:

- Make the password at least six characters.
- Make sure the password doesn't spell a word.
- Make sure the password doesn't consist of publicly obtainable information about you; for example, the initials and birth date for you, your spouse, or children.
- Include at least two, nonconsecutive numbers in the password.

Type the password twice to ensure that you have typed it correctly.

If you must record the password, make sure it is stored in a well-secured place.

The key ring password must be specified when the server is started. Check **Automatic login** if you want the server to stash the password and specify it for you whenever the server is started. To stash the password, the server encrypts the key ring password and puts it in a file that has the same name as the key ring file, except the extension is `.sth`.

Stashing your password is less secure so it's generally a good practice not to stash it. However, there may be cases where you need to stash the password; for example, if you have a remote server and you want it to automatically restart after a power failure.

Next, to request the server's certificate, fill in the rest of the form.

5. Specify Distinguished Name

Specify the Distinguished Name you want associated with the server's certificate and used to identify the server's public key. In its simplest form, a Distinguished Name is someone's name.

The information you provide for Distinguished Name depends upon whether you want a:

- VeriSign Low Assurance Certificate. See “Specify Distinguished Name for low assurance certificate” on page 261.
- VeriSign Secure Server Certificate or Other Certificate. See “Specify Distinguished Name for secure server or other certificate” on page 262.

Specify Distinguished Name for low assurance certificate: For this certificate, the Distinguished Name is the full name of the user for whom the certificate is being requested.

Distinguished Name

A Distinguished Name is a unique name that is associated with the certificate and public key. For a VeriSign low assurance certificate, the Distinguished Name is the full name of the user for whom the certificate is being requested. If you request more than one certificate for a user, each request must have a unique name for **User's full name**. For example, you might specify Mary Ann Jones1 for the user's first certificate request and Mary Ann Jones2 for the user's second certificate request.

User's e-mail address should contain the address where you want the CA to mail the certificate.

User's full name

User's e-mail address

For **User's full name**, specify the user's full name and make sure it is unique. For example, if you request more than one certificate for Mary Ann Jones, you might specify Mary Ann Jones1 for the first request and Mary Ann Jones2 for the second request.

User's e-mail address should contain the user's address where you want VeriSign to mail the certificate.

Next, you need to provide mailing information, as described in “6. Specify mailing option” on page 263.

Using SSL

Specify Distinguished Name for secure server or other certificate: For these certificates, the Distinguished Name is the server name and the location of the server.

Distinguished Name

The Distinguished Name is a unique name that is associated with the certificate and public key. For this certificate, the Distinguished Name is the Server name and the location of the server. Server name is the X.500 common name. It is usually the fully qualified TCP/IP host name.

| | | |
|---------------------|----------------------|----------------------------|
| Server name | <input type="text"/> | |
| Organizational unit | <input type="text"/> | (optional) |
| Organization | <input type="text"/> | |
| Locality/City | <input type="text"/> | (optional) |
| State/Province | <input type="text"/> | (minimum three characters) |
| Postal code | <input type="text"/> | |
| Country | <input type="text"/> | |

User's e-mail address should contain the address where you want the CA to mail the certificate.

User's e-mail address

Note: The text for Distinguished Name will vary, depending on whether you're on the VeriSign Secure Server Certificate form or the Other Certificate form. Also, the User's e-mail address field, which is on the Other Certificate form, is not on the VeriSign Secure Server Certificate form.

You may use non-alphanumeric characters in Distinguished Name; however, keep in mind that some platforms have special uses for some of these characters. Provide the following information for Distinguished Name:

- **Server Name:** The X.500 common name of the server. Usually, it is the TCP/IP fully qualified host name; for example, www.ibm.com. For a VeriSign secure server certificate request, it must be the fully qualified host name.
- **Organizational Unit:** Optionally, the organizational unit. For example, you might specify the division of your company.
- **Organization:** The name of your organization, such as the name of your company. For a secure server certificate, if you already have an account with VeriSign, the name in this field should match the name on that account.
- **Locality/City:** Optionally, the city or locality where your server resides.
- **State/Province:** The state or province where the server resides. You must specify at least **three** characters.
- **Postal Code:** The postal code or zip code where the server resides.
- **Country:** The two-character country code for the country in which the server resides.

On the Other Certificate form, **User's e-mail address** should contain the user's address where you want the CA to mail the certificate. If you're requesting this server's certificate, which you plan to process as a CA, leave this field blank.

6. Specify mailing option

The mailing option depends on the kind of certificate being requested:

- For a VeriSign low assurance certificate, see “Mailing option for low assurance certificate” on page 263.
- For a VeriSign secure server certificate, see “Mailing option for secure server certificate” on page 263.
- For an Other certificate, see “Mailing option for other certificate” on page 263.

Mailing option for low assurance certificate: A VeriSign low assurance certificate request must be electronically mailed to VeriSign. If you are behind a firewall, verify with your system administrator what you need to do to electronically mail this request.

Mail To

A low assurance certificate request must be electronically mailed to VeriSign. If you are behind a firewall, you may need to talk to your system administrator about how to electronically mail this request.

Mail to

Go to “7. Save copy of certificate request” on page 264 for a description of what you need to do next.

Mailing option for secure server certificate: There is no mailing option on the form for a VeriSign secure server certificate. A secure server certificate request must be saved in a file and manually mailed to VeriSign. The VeriSign home page, <https://www.verisign.com/enroll.s>, has instructions on what you need to provide in order to obtain a secure server certificate from VeriSign. Some e-mail programs may alter files and should not be used to send a certificate request. For example, some programs may pad the lines of a certificate request with blanks and invalidate it. Also, if you are behind a firewall, verify with your system administrator what you need to do to electronically mail this request.

Go to “7. Save copy of certificate request” on page 264 for a description of what you need to do next.

Mailing option for other certificate: For an Other certificate, you need to consult with your CA to determine how to mail the certificate request.

Using SSL

Mail To

Choose the mailing option recommended by your CA. If you are acting as your own CA to request your CA certificate or to request this server's certificate that you plan to process as CA, choose **Don't mail**.

Mail to

Don't mail

After consulting with your CA, choose one:

- **Mail to:** Choose this option and enter the electronic address of your CA if you want to electronically mail the certificate request. If you are behind a firewall, verify with your system administrator what you need to do to electronically mail this request.
- **Don't mail:** Choose this option if you plan to act as your own CA for this server's certificate or if you don't want to electronically mail the certificate request to another CA at this time. You can manually mail the certificate request later; however, some e-mail programs may alter files and should not be used to send a certificate request. For example, some programs may pad the lines of a certificate request with blanks and invalidate it. Also, if you are behind a firewall, verify with your system administrator what you need to do to electronically mail this request.

7. Save copy of certificate request

It's a good idea to save the certificate request in a file. If you're requesting a VeriSign secure server certificate, you must save the request to file and manually mail it later. If you're acting as your own CA for this server's certificate, you also must save the certificate request in a file. (After you complete this form, you can use the instructions in Chapter 11, "Acting as a certification authority for a private Web network" on page 269 to become a CA and to process this server's certificate.)

Save Copy

Specify a unique, fully qualified path and file name for the file where you want to save the certificate request. You need to manually mail the certificate request to VeriSign.

Save certificate request to file

Note: The text on this form for Save Copy will vary, depending on whether you're on the VeriSign low assurance, VeriSign secure server, or Other certificate form.

Specify a unique fully qualified path and file name for the file where you want to keep the certificate request. If you specify a file that doesn't exist, the server creates the file for you.

8. Process your request

Check your form and make sure you entered each item correctly. Click **Apply** to process the form. You receive a Confirmation that the form was successfully processed.

The server:

- Stores the key pair in an encrypted format on the key ring and adds the default trusted roots that are shipped with the product to the key ring.
- If you specified **Automatic login**, encrypts the key ring password and puts it in a file that has the same name as the key ring file, except the extension is .sth.
- Creates the certificate request and stores it in the file you specified and:
 - For a low assurance certificate, the certificate request is electronically mailed to VeriSign.
 - For a secure server certificate, you have to manually mail the certificate request to VeriSign. Go to the VeriSign home page, <https://www.verisign.com/enroll.s>, for instructions on what you need to provide in order to obtain a secure server certificate from VeriSign.
 - For an other certificate, if you're acting as your own CA for this server's certificate, you need to process the certificate request. Go to Chapter 11, "Acting as a certification authority for a private Web network" on page 269 for instructions on how to become a CA and for how to process this server's certificate. For any other CA, follow its instructions.

The CA processes your certificate request. When the CA is satisfied that you have met all of its requirements, it electronically mails a certificate to you. Next, you need to receive the server's certificate into the key ring that contains the public-private key pair.

Fill in Receive Certificate Form

Use the Receive Certificate form to receive into the key ring the certificate electronically mailed to you by your CA.

Using SSL

Receive Certificate

Use this form to receive a certificate into its key ring after it has been processed by a certification authority (CA). This form can also be used to create a signed certificate for you to use as a CA for a private Web network.

Specify the unique, fully qualified path and file name for the file that contains the certificate you are receiving. Specify the fully qualified path and file name for the key ring where the certificate will be kept. Specify the key ring password.

| | |
|-------------------------------------|---|
| Name of file containing certificate | <input type="text" value="D:/WWW/BIN/Cert.txt"/> |
| Key ring | <input type="text" value="D:/WWW/BIN/keyfile.kyr"/> |
| Key ring password | <input type="text"/> |

1. Specify Receive Certificate information

Before filling in this form, use your e-mail program to receive the certificate into a unique file. Some e-mail programs may alter files and should not be used to receive certificates. For example, some programs pad the lines of the certificate with trailing blanks, which invalidates it.

For the Receive Certificate form, provide:

- **Name of file containing certificate:** The fully qualified path and file name of the file that contains the certificate.
- **Key ring:** The fully qualified path and file name of the key ring. This is the key ring you specified on the Create Key and Request Certificate form when you created the server's public-private key pair and the request for this certificate.
- **Key ring password:** The password for the key ring. This is the password you specified on the Create Key and Request Certificate form when you created the server's public-private key pair and the request for this certificate.

2. Process your request

Check your form and make sure you entered each item correctly. Click **Apply** to process the form.

You receive a Confirmation that the form was successfully processed.

The server stores the certificate in the key ring. It is referenced in the key ring with the **key name** you specified on the Create Key and Request Certificate form.

After you receive your Confirmation for the Receive Certificate form, you can make changes to other Configuration and Administration forms. However, when you are ready for the changes you've made to any security forms to take effect, you must stop the server and then start it again. The server will not pick up your changes if you only restart it.

When you start the server, you will be prompted for the key ring password unless you stashed it by selecting **Automatic login** on the Create Key and Request Certificate form.

Step 2. Specifying SSL client authentication in protection setups and ACL files

With SSL V3, you can set up your server for SSL client authentication. The server requests a certificate from any client making an **https** request. The server establishes a secure connection whether or not the client has a valid certificate.

You can restrict who can access documents by using password files and/or user or group authentication in protection setups as described in Chapter 7, "Protecting your server" on page 201. You can further restrict who can access documents by coding SSL client authentication parameters on protection setups, ACL files, or both. See "Creating protection setups for SSL client authentication" on page 212 for more information on coding protection setups. See "Step 5. Limiting access to individual files" on page 214 for more information on coding ACL files.

Step 3. Using SSL with your server

Once you have a key pair and a signed certificate, you can begin serving SSL protected documents to SSL browsers.

Before you begin using SSL, you should consider doing the following:

- Set up a proxy server. See "Running your server as a proxy" on page 11 for information on how to set up your server as a proxy.
- If you are writing CGI programs, make use of the following SSL environment variables:
 - HTTPS
 - HTTP_KEYSIZE

Chapter 11. Acting as a certification authority for a private Web network

To conduct commercial business on the Internet, you will want to get a secure server certificate from a widely known certification authority (CA), such as VeriSign.

You may, however, have a project for your company, university, or group where you want to act as your own CA for a private Web network. The Internet Connection Secure Server allows you to be your own CA. As a CA, you will use a utility called certutil, which is provided with the Internet Connection Secure Server, to create signed certificates. However, clients must have browsers, such as Secure WebExplorer or Netscape Navigator, that can receive your CA certificate and designate it as a trusted root.

Note: Your use of this utility 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 utility for any other purpose.

This chapter tells you how to become your own CA and then how to process certificate requests as a CA.

| | |
|--|-----|
| Becoming a CA | 269 |
| Create CA's public-private key pair and request CA's certificate | 270 |
| Receive CA's certificate | 273 |
| Designate the CA key as a trusted root | 274 |
| Processing certificates as a CA | 276 |
| Use certutil command to process client and server certificates | 277 |

Becoming a CA

Your job as CA is to verify that a certificate should be issued for a client or server. You need to make sure that the person making the request has a legitimate claim to the name in the request. For example, if your company is working on a top secret project, your issuance of a certificate can give a person access to top secret information.

After you have verified a person's claim, you will use the certutil command to process the certificate request. The output from the certutil command is a certificate signed with your CA private key.

After you verify that the information in the certificate created by certutil is correct, you will:

- Send your CA certificate to the client or server for whom you're acting as CA. The client or server needs to receive your CA certificate and designate it as a trusted root.
- Send the client or server its certificate, which you processed as a CA. The client or server needs to receive its certificate into its key ring after it has received the CA certificate and designated it as a trusted root.

Before you can process certificate requests as a CA, you need to do the following to become a CA:

- Create your CA public-private key pair and request a CA certificate
- Receive your CA certificate (which is the same as the CA certificate request) into your CA key ring
- Designate your CA key as a trusted root

You must carefully protect your CA private key. If it is compromised, then all the certificates you've issued will also be compromised.

Create CA's public-private key pair and request CA's certificate

This chapter describes how to use the Create Key and Request Certificate form to create your CA public-private key pair and request your CA certificate.

Create Key and Request Certificate

Choose the certification authority (CA) from whom you want to obtain a certificate. VeriSign is a widely known CA. For information about obtaining a certificate from VeriSign, you can access the [VeriSign home page](#). If you want to use another CA or to act as your own CA for a private Web network, choose Other.

VeriSign (Low Assurance Certificate)
 VeriSign (Secure Server Certificate)
 Other

- 1 Choose **Other**.
- 2 Click **Apply** to process this part of the Create Key and Request Certificate form. The Other Certificate form appears.
- 3 Use **Create Key** to create your CA key pair.

Create Key

Specify a unique, meaningful name, which will be used to identify the public-private key pair. Also specify the size of the key pair and the fully qualified path and file name for the key ring where the key pair will be kept. If you are creating your CA keys, you should keep them in a unique key ring.

Key name Size bits
Key ring

Specify Key name: Specify a meaningful, unique name to identify your CA key pair. The key name is the label that identifies the key pair and certificate in the key ring. You may use non-alphanumeric characters in key names; however, keep in mind that some platforms have special uses for some of these characters.

Specify Key ring: Specify the fully qualified path and file name for the key ring file. It's best to use a separate key ring to keep your CA key pair. If you specify a file that doesn't exist for key ring, the server creates the file for you.

Specify the Size of the Key Pair: Your private CA key will be used to sign the server and client certificates you process as CA. Your CA public key will be used by those clients and servers to verify your CA signature. The larger the CA key pair, the more secure the key pair will be. If the CA private key is compromised, then all the certificates issued by the CA are also compromised.

- 4 Specify a Password:** Use this part of the form to specify the password that protects your CA key ring. You should keep your CA keys in a separate key ring with its own password. If your CA private key is compromised, then all the certificates you've issued as a CA are also compromised.

The password must be from the U.S. English character set.

Key Ring Password

Specify a password for the key ring. The key ring password must be specified each time the server is started. If you check **Automatic login**, the password is automatically specified when the server is started. If you are specifying the password for the server's key ring, make sure this box is checked if you want non-interactive startup. If you are specifying the password for your CA key ring, make sure this box is not checked. If your CA keys are compromised, all the certificates you have issued are also compromised.

Password
Password (for verification)
 Automatic login

The security of your CA private key, which is used to sign the client and server certificates you process as a CA, depends upon this password. Here are some commonly accepted rules for specifying passwords:

- Make the password at least six characters.
- Make sure the password doesn't spell a word.

- Make sure the password doesn't consist of publicly obtainable information about you; for example, the initials and birth date for you, your spouse, or children.
- Include at least two, nonconsecutive numbers in the password.

Type the password twice to ensure that you have typed it correctly.

If you must record the password, make sure it is stored in a well-secured place.

If, as recommended, you put your CA key pair on a separate key ring, do **not** check **Automatic Login** to stash the password for the CA key ring.

- 5 Specify Distinguished Name:** Specify the Distinguished Name you want associated with your CA certificate and used to identify your CA public key.

Distinguished Name

The Distinguished Name is a unique name that is associated with the certificate and public key. For this certificate, the Distinguished Name is the Server name and the location of the server. Server name is the X.500 common name. It is usually the fully qualified TCP/IP host name.

| | | |
|---------------------|----------------------|----------------------------|
| Server name | <input type="text"/> | |
| Organizational unit | <input type="text"/> | (optional) |
| Organization | <input type="text"/> | |
| Locality/City | <input type="text"/> | (optional) |
| State/Province | <input type="text"/> | (minimum three characters) |
| Postal code | <input type="text"/> | |
| Country | <input type="text"/> | |

User's e-mail address should contain the address where you want the CA to mail the certificate.

User's e-mail address

Provide the following information for Distinguished Name:

- Server Name: Usually the X.509 common name of the server, which is usually the TCP/IP fully qualified host name; for example, www.ibm.com.
- Optionally, Organizational Unit
- Organization
- Optionally, Locality or City
- At least **three** characters for State or Province
- Postal code or zip code.
- The two-character country code for the country in which the server resides

- 6 Don't specify User's e-mail address.**
- 7 Specify Mailing Option:** Choose **Don't mail**.

Mail To

Choose the mailing option recommended by your CA. If you are acting as your own CA to request your CA certificate or to request this server's certificate that you plan to process as CA, choose **Don't mail**.

- Mail to
- Don't mail

- 8 Save Your Certificate Request:** Save your CA certificate request in a unique file. The server creates the file for you.

Save Copy

Specify a unique, fully qualified path and file name for the file where you want to save the certificate request.

Save certificate request to file

- 9** Check your form and make sure you entered each item correctly.
- 10** Click **Apply** to process the form. You receive a Confirmation that the form was successfully processed.
- The server:
- Stores the key pair in an encrypted format on the key ring.
 - Creates the certificate request and stores it in the file you specified.
- Next, you need to receive your CA certificate request.

Receive CA's certificate

Use the Receive Certificate form to receive the CA certificate request that you saved. The CA's certificate is the same as the certificate request.

Receive Certificate

Use this form to receive a certificate into its key ring after it has been processed by a certification authority (CA). This form can also be used to create a signed certificate for you to use as a CA for a private Web network.

Specify the unique, fully qualified path and file name for the file that contains the certificate you are receiving. Specify the fully qualified path and file name for the key ring where the certificate will be kept. Specify the key ring password.

| | |
|-------------------------------------|--------------------------|
| Name of file containing certificate | D:/WWW/BIN/caCertReq.txt |
| Key ring | D:/WWW/BIN/cakeyfil.kyr |
| Key ring password | |

1 Provide:

Name of file containing certificate: The fully qualified path and file name of the file in which you saved the CA certificate request (**Save certificate request to file** on the Create Key and Request Certificate form).

Key ring: The fully qualified path and file name of the key ring you specified on the Create Key and Request Certificate form for your CA certificate.

Key ring password: The password you specified on the Create Key and Request Certificate form for your CA key ring.

2 Check your form and make sure you entered each item correctly.

3 Click **Apply** to process the form. You receive a Confirmation that the form was successfully processed.

The server stores your CA certificate in the CA key ring. It is referenced in the key ring with the **key name** you specified on the Create Key and Request Certificate form.

You now have your CA certificate (also known as a self-signed certificate) signed with your CA private key in your CA key ring.

Next, the CA key needs to be a trusted root on the server.

Designate the CA key as a trusted root

Use the Key Management form.

Key Management

Use the key management forms to manage your keys and certificates. This form shows the current key ring that you'll be working with.

Current key ring: D:/WWW/BIN/cakeyfil.kyr

Specify the key ring password.

Key Ring Password

Choose the key management task you want to perform for the current key ring.

- Change Password - Change key ring password
- Manage Keys - Delete keys or make a key the default in this key ring
- Export Keys - Transfer key pair or certificate to another key ring or computer
- Import Keys - Transfer key pair or certificate to this key ring
- Request Certificate - Request certificate for an existing key
- Designate Trusted Root Keys - Designate keys as trusted root keys
- Remove Trusted Root Keys - Remove trusted root key designation

- 1 Check the key ring file name shown on this form. If it is not the key ring that contains your CA key, go to the Security Configuration form and designate your CA key ring as the current key ring. After you receive your confirmation message, come back to the Key Management form, which should show the CA key ring as the current key ring.
- 2 Specify the key ring password for the CA key.
- 3 Choose the option **Designate Trusted Root Keys**.
- 4 Click **Apply** to process the form. The Designate Trusted Root Keys form appears.

Designate Trusted Root Keys

Use this form to designate a key in the current key ring as a trusted root key.

Current key ring: D:/WWW/BIN/cakeyfil.kyr

Choose the key you want to designate as a trusted root. Only the public key of a certification authority should be designated as a trusted root.

Keys

cakey 

On the Designate Trusted Root Keys form:

- 1 Select the name of the CA key.

- 2 Click **Apply** to process the form. You receive a Confirmation that the form was successfully processed.
- 3 If you have a key ring that contains keys the server uses for secure network communications, it should be the current key ring. You can use the Security Configuration form to designate the server's key ring as the current key ring.

Now, you are ready to process the certificate request for clients and servers.

Processing certificates as a CA

After you have created your CA key pair, obtained your CA certificate, and designated your CA public key as a trusted root, you are ready to act as a CA and process certificates for this server and for other servers and clients.

Servers can use the Configuration and Administration forms to create their public-private key pair and request their certificate. They should use the Other Certificate form. If you provide them with your e-mail address, they can specify it on the form and have the certificate request electronically mailed to you. Or, they can save the certificate request in a file and electronically mail it to you later.

Some browsers, such as Secure WebExplorer and Netscape Navigator, allow clients to create their public-private key pair and request their certificate. Clients can electronically mail their certificate requests to you.

Some e-mail programs may alter files and should not be used to send or receive certificate requests or certificates. For example, some programs may pad the lines of the certificate request or certificate with blanks and invalidate it.

To process certificates for clients and servers, you will:

- Receive certificate requests from clients and servers and verify that the person making the request has a legitimate claim to the name in the request.
- Use the `certutil` command to process the certificate request. The output from the `certutil` command is a certificate signed with your CA private key.
- Check the information printed in the output file and verify to whom the certificate is being issued before sending it to the client or server.
- Send the client or server:
 - Your CA certificate. This certificate must be received into the key ring of the client or server and then designated as a trusted root.
 - The client or server certificate that was processed by you as CA. After the client or server has received the CA certificate into its key ring and designated

it as a trusted root, the client or server can receive its certificate into the key ring.

Then, the client or server can use the certificate you created as a CA to communicate securely with other clients and servers within your private Web network.

Use certutil command to process client and server certificates

Use the certutil command to process certificate requests for the clients and servers for whom you're acting as CA. The output of the certutil command is a client or server certificate that has been signed with your private CA key.

Before using the certutil command, verify that the person making the request has a valid claim to the name in the request.

Format:

```
certutil [-p validity-period] [-k ca-key-ring] < cert-req-file > cert-file-name
```

- *validity-period* can be 1 to 9999 for the number of days the certificate is valid. If you don't specify *validity-period*, the default is 365 days.
- *ca-key-ring* is the fully qualified path and file name you specified for the CA **Key ring** on the Create Key and Request Certificate form. If you don't specify *ca-key-ring*, the default is keyfile.kyr.
- *cert-req-file* is the fully qualified path and file name of the file that contains the client or server certificate you are processing. For clients or servers who mail their certificate request to you through e-mail, it is the file into which you received the certificate request with your e-mail program. For a certificate for this server, it is the file you specified for **Save certificate request to file** on the Create Key and Request Certificate form when you were requesting the server certificate.
- *cert-file-name* is the unique, fully qualified path and file name of the output file where you want to put the signed certificate created by the certutil command.

Example:

```
certutil -p 730 -k D:/WWW/BIN/cakeyfil.kyr < D:/WWW/BIN/CertReq.txt  
> D:/WWW/BIN/Cert.txt
```

Note: The certutil command should be on one line. It is shown on more than one line because of formatting.

After you enter the command, you are prompted for the password of the key ring that has the CA key.

If certutil cannot create the signed certificate, you will get an error message.

It's a good idea to check the output file and verify that the certificate correctly states to whom it is being issued.

What you do next depends on whose certificate you are processing:

- If you are processing this server's certificate, go to "Receive certificate for this server" on page 278.
- If you are processing certificates for other servers or clients, go to "Send clients and other servers their processed certificates" on page 280.

Receive certificate for this server

If you are processing this server's certificate, you need to:

- 1 Use the Receive Certificate form to receive the CA certificate into the server's key ring.

Receive Certificate

Use this form to receive a certificate into its key ring after it has been processed by a certification authority (CA). This form can also be used to create a signed certificate for you to use as a CA for internal communications.

Specify the unique, fully qualified path and file name for the file that contains the certificate you are receiving. Specify the fully qualified path and file name for the key ring where the certificate will be kept. Specify the key ring password.

| | |
|-------------------------------------|---|
| Name of file containing certificate | <input type="text" value="D:/WWW/BIN/caCertReq.txt"/> |
| Key ring | <input type="text" value="D:/WWW/BIN/keyfile.kyr"/> |
| Key ring password | <input type="text"/> |

Provide:

- **Name of file containing certificate:** Specify the file name of the file into which you saved the CA certificate request. The request is the same as the CA's certificate.
 - **Key ring:** The key ring you specified on the Create Key and Request Certificate form when you created this server's keys and requested its certificate.
 - **Key ring password:** The password you specified on the Create Key and Request Certificate form for this server's key ring.
- 2 Check your form and make sure you entered each item correctly.
 - 3 Click **Apply** to process the form. You receive a Confirmation that the form was successfully processed.

The server stores the certificate in its key ring. It is referenced in the key ring with the **key name** you specified on the Create Key and Request Certificate form for the server's certificate.

- 4 Go to the Key Management form. Make sure that the server's key ring is the current key ring. If it is not the current key ring, go to the Security Configuration form and make the server's key ring the current key ring. After you receive the confirmation message, come back to the Key Management form.
- 5 On the Key Management form, specify the key ring password for the server's key ring.
- 6 Choose the option **Designate Trusted Root Keys** and click **Apply** to process the form. The Designate Trusted Root Keys form appears.
- 7 On the Designate Trusted Root Keys form, select the CA certificate. For the CA certificate, you will see the Distinguished Name (DN) of the CA rather than the key name.

Designate Trusted Root Keys

Use this form to designate a key in the current key ring as a trusted root key.

Current key ring: D:/WWW/BIN/keyfile.kyr

Choose the key you want to designate as a trusted root. Only the public key of a certification authority should be designated as a trusted root.

Keys

PC = 29837, CN = Your Private Web CA, OU = Your Organizational Unit Name, O = Yc
mykey

- 8 Click **Apply** to process the form. You receive a Confirmation that the form was successfully processed.
- 9 Use the Receive Certificate form to receive the server's certificate into its key ring.

Receive Certificate

Use this form to receive a certificate into its key ring after it has been processed by a certification authority (CA). This form can also be used to create a signed certificate for you to use as a CA for a private Web network.

Specify the unique, fully qualified path and file name for the file that contains the certificate you are receiving. Specify the fully qualified path and file name for the key ring where the certificate will be kept. Specify the key ring password.

| | |
|-------------------------------------|---|
| Name of file containing certificate | <input type="text" value="D:/WWW/BIN/Cert.txt"/> |
| Key ring | <input type="text" value="D:/WWW/BIN/keyfile.kyr"/> |
| Key ring password | <input type="text"/> |

Provide:

- **Name of file containing certificate:** Specify the file name you provided for *cert-file-name* when you used the *certutil* command.
- **Key ring:** The key ring you specified on the Create Key and Request Certificate form when you created this server's keys and requested its certificate.
- **Key ring password:** The password you specified on the Create Key and Request Certificate form for this server's key ring.

10 Check your form and make sure you entered each item correctly.

11 Click **Apply** to process the form. You receive a Confirmation that the form was successfully processed.

The server stores the certificate in its key ring. It is referenced in the key ring with the **key name** you specified on the Create Key and Request Certificate form for the server's certificate.

After you receive your Confirmation for the Receive Certificate form, you can make changes to other Configuration and Administration forms. However, when you are ready to use the server's keys for secure network communications, you must stop the server and then start it again. The server will not pick up the changes you've made to the key ring if you only restart it.

Send clients and other servers their processed certificates

You need to send the clients and other servers for whom you are acting as CA:

- Your CA certificate. Clients and other servers need to receive this certificate and designate it as a trusted root. You can send clients and servers your CA certificate request. It is your signed CA certificate.
- The client or server certificate you processed as CA. After the client or server has received the CA certificate and designated it as a trusted root, the client or server can receive its certificate into the key ring.

Note: Some e-mail programs may alter files and should not be used to send or receive certificate requests or certificates. For example, some programs may pad the lines of the certificate request or certificate with blanks and invalidate it.

After a client or server receives the CA certificate, designates it as a trusted root, and receives the client or server certificate, the client or server certificate can be used for secure network communications. Before a server's keys are used for secure network communications, the server must be stopped and then started again. The server will not pick up the changes you made to the server's key ring if you only restart it.

Chapter 12. Supported key lengths and encryption modes

The U.S. regulates products used for encryption and prohibits their export unless their key size is strictly limited. This chapter summarizes the key sizes and the SSL encryption modes for U.S. and export products. These key sizes and encryption modes are dictated by U.S. export rules.

Customers in the U.S. and Canada can install the U.S.-Canadian version or the export version of the Internet Connection Secure Server.

| | |
|-----------------------------------|-----|
| Public and private keys | 283 |
| SSL encryption modes | 284 |

Public and private keys

Public and private keys are used to encrypt and decrypt messages, data, and message digests. They are also used for creating message digests as part of digital signature.

Key sizes for U.S.-Canadian version: The U.S.-Canadian version of the server can do the following:

- Generate keys from 508-1024 bits
- Encrypt data with keys from 508-1024 bits
- Sign data with keys from 508-1024 bits
- Check signatures with keys from 508-1024 bits

Key sizes for export version: The export version of the server can do the following:

- Generate keys from 508-512 bits
- Encrypt data with keys from 508-512 bits
- Sign data with keys from 508-1024 bits
- Check signatures with keys from 508-1024 bits

As US export laws are updated, the supported key lengths and algorithms are subject to change. Refer to the Internet Connection Family web site at <http://ics.raleigh.ibm.com> for the latest information.

SSL encryption modes

SSL uses a security handshake to initiate the TCP/IP connection between the client and the server. During the handshake, the client and server agree on the level of security they will use, and the client authenticates the server. After that, SSL is used to encrypt and decrypt the information in both the request and the server response.

SSL modes for U.S.-Canadian version

- RC4 128 bit
- RC2 128 bit
- DES 56 bit
- Triple DES (EDE) 192 bit

SSL Modes for export version

- RC4 export (40 bit)
- RC2 export (40 bit)
- DES 56 bit

Part 4. Appendixes

Appendix A. Command reference

This chapter describes how to use the server's commands.

| | |
|------------------|-----|
| certutil command | 287 |
| cgiparse command | 288 |
| cgiutils command | 291 |
| htadm command | 293 |
| htimage command | 295 |
| httpd command | 298 |

certutil command

For a secure server, use the certutil command to process certificate requests for the clients and servers for whom you're acting as a Certificate Authority (CA). The output of the certutil command is a client or server certificate that has been signed with your private CA key.

Note: How to become your own CA and how to process certificate requests as a CA is described in Chapter 11, "Acting as a certification authority for a private Web network" on page 269. You should follow the step-by-step instructions in that chapter unless you are familiar with the procedure.

Syntax

```
certutil [-p validity-period] [-k ca-key-ring]  
< cert-req-file > cert-file-name
```

Note: The certutil command should be on one line. It is shown on more than one line because of formatting.

The < and > symbols, redirect the standard input and standard output.

Flags

-p *validity-period*

Validity period in days of the certificate to be issued. Acceptable values are from 1 to 9999. If you omit this flag, the default validity period is 365 days.

Commands

-k *ca-key-ring*

Specifies the fully qualified path and file name of the key ring file containing your private CA key. This is the name you specified as the CA **Key ring** on the Create Key and Request Certificate form, if you did not accept the default key ring file, keyfile.kyr. If you omit this flag, the default keyfile.kyr is assumed.

cert-req-file

Specifies the fully qualified path and file name of the file that contains the client or server certificate you are processing.

For clients or servers who mail their certificate request to you through e-mail, it is the file into which you received the certificate request with your e-mail program.

For a certificate request on the same server, it is the file you specified for **Save certificate request to file** on the Create Key and Request Certificate form when you were requesting the server certificate.

cert-file-name

Specifies the unique fully qualified path and file name of the output file where you want to put the certificate created by this command.

Example

To create a certificate that is valid for two years:

```
certutil -p 730 -k D:/WWW/BIN/akeyfil.kyr < D:/WWW/BIN/CertReq.txt  
> D:/WWW/BIN/Cert.txt
```

Note: The certutil command must be on one line. It is shown here on more than one line because of formatting restrictions.

After you enter the command, you are prompted for the key ring password. This is the password for the key ring that holds the CA key.

If the certutil command cannot create the signed certificate, it returns an error message.

cgiparse command

Use the cgiparse command to parse the QUERY_STRING environment variable for CGI scripts. If the QUERY_STRING environment variable is not set, the command reads CONTENT_LENGTH characters from its standard input. All returned output is written to its standard output.

This corresponds to the GET and POST methods from HTML forms. QUERY_STRING and CONTENT_LENGTH are environment variables. For a GET request,

Commands

QUERY_STRING holds the form data. For a POST request, CONTENT_LENGTH is set and the form data is sent on standard input.

Syntax

```
cgiparse -Flag [Modifier]
```

Flags

Flags have one-character equivalents: -k -f -v -r -i -s -p -c -q -P

-keywords

Parses QUERY_STRING for keywords. Keywords are decoded and written to standard output, one per line.

-form

Parses input as form request. Outputs one or more SET statements that set environment variables beginning with FORM_ followed by a field name. Field values are the contents of the variables.

-value *field-name*

Parses input as form request. Prints only the value of *field-name*.

-read

Reads CONTENT_LENGTH characters from standard input and writes them to standard output.

-init

If QUERY_STRING is not set, reads the value of standard input and returns a SET statement that sets QUERY_STRING to this value. This can be used with both the GET and POST methods. A typical use is:

```
'cgiparse -init > tempfile'  
'CALL tempfile'
```

This will set the QUERY_STRING environment variable, regardless of whether the GET or POST method was used.

The cgiparse command may be called multiple times in the same script when the GET method is used, but it should only be called once if the POST method is used. With the POST method, after standard input is read, the next cgiparse would find standard input empty and would hang.

-sep *separator*

Specifies the string used to separate multiple values. If you are using the **-value** flag, the default separator is newline. If you are using the **-form** flag, the default separator is a comma (,).

-prefix *prefix*

Used with **-POST** and **-form**, specifies the prefix to use when creating environment variable names. The default is "FORM_".

Commands

-count

Used with **-keywords**, **-form**, and **-value**, returns a count of items related to these flags.

-keywords

Returns the number of keywords

-form

Returns the number of unique fields (multiple values are counted as one)

-value *field-name*

Returns the number of values for *field-name* (if there is not a field named *field-name*, output is 0).

-number

Used with **-keywords**, **-form**, and **-value**, returns the specified occurrence related to these flags.

-keywords

Returns the *n*'th keyword. (For example **-2 -keywords** outputs the second keyword.)

-form

Returns all the values of the *n*'th field. (For example **-2 -form** outputs all the values of the second field.)

-value *field-name*

Returns the *n*'th of the multiple values of field *field-name*. (For example **-2 -value -whatsit** outputs the second value of the **whatsit** field).

-quiet

Suppresses all error messages. (Non-zero exit status still indicates error.)

-POST

Information from stdin is directly decoded and parsed into shell variables, QUERY_STRING is not used.

Exit statuses

- 0 Success
- 1 Illegal command line
- 2 Environment variables not set correctly
- 3 Failed to get requested information (for example, there is no such field or QUERY_STRING contains keywords when form field values are requested).

Note: When you receive one of these error codes, you may receive additional informational messages, too. The message varies depending on the command issued.

cgiutils command

Use the `cgiutils` command in no-parse header programs to produce a full HTTP 1.0 response.

Note: If you want to supply your own no-parse header (`nph`) programs specifically to return your own return values, the name of the program must begin with **nph-**. This prevents the server header from overriding your return value with the standard server return value.

Syntax

```
cgiutils -Flag [Modifier]
```

If *Modifier* contains blanks, enclose it in quotes.

Flags

-version

Returns version information.

-nodate

Does not return the Date: header.

-noel

Does not return a blank line after headers. This is useful if you want other MIME headers after the initial header lines.

-status *nnn*

Returns full HTTP response with status code *nnn*, instead of only a set of HTTP headers. Do not use this flag if you only want the Expires: header.

-reason *explanation*

Specifies the reason line for the HTTP response. You can only use this flag with the **-status** *nnn* flag. If the explanation text contains more than one word, you must enclose it in quotes.

-ct [*type/subtype*]

Specifies MIME Content-Type header. This example specifies a MIME content type of text/html:

```
cgiutils -ct text/html
```

If you omit the *type/subtype*, the MIME content type is set to the default text/plain. This example sets the MIME content type to text/plain.

```
cgiutils -ct
```

Commands

-ce *encoding*

Specifies MIME Content-Encoding header. For example:

```
cgiutils -ce x-compress
```

-cl *language-code*

Specifies MIME Content-Language header. For example:

```
cgiutils -cl en_UK
```

-length *nnn*

Specifies MIME Content-Length header.

-expires *Time-Spec*

Specifies MIME Expires header. This flag specifies the time to live (the expiration date of a document) in any combination of years, months, weeks, days, hours, minutes, and seconds. These values and keywords specify the length of time a document is considered valid. You must enclose them in quotes. For example:

```
cgiutils -expires "2 days 12 hours"
```

The `cgiutils` command adds the time you specify to the current Greenwich Mean Time to determine the expiration date. The expiration date is put in the Expires: header in the HTTP format.

-expires now

Produces an Expires: header that matches the Date: header.

-uri *URI*

Specifies the Universal Resource Identifier (URI) for the returned document. URI can be considered to be the same as URL.

-extra *xxx: yyy*

Specifies an extra header that cannot otherwise be specified for `cgiutils`.

Examples

- This example calculates the expiration date for the Expires: header.

```
cgiutils -expires "1 year 3 months 2 weeks 4 days 12 hours 30 mins"
```
- The following example specifies a status code and reason, and sets the Expire: header equal to the Date: header.

```
cgiutils -status 200 -reason "Virtual doc follows" -expires now
```

This might produce headers similar to these:

```
HTTP/1.0 200 Virtual doc follows
MIME-Version: 1.0
Server: IBM-ICS
Date: Tuesday, 05-Jan-96 03:43:46 GMT
Expires: Tuesday, 05-Jan-96 03:43:46 GMT
```

Commands

The `cgiutils` command automatically produces the `Server:` header because it is available in the CGI environment. The `Date:` field is also automatically generated unless the `-nodate` flag is specified.

This would include a blank line after the output to mark the end of the MIME header section. If you want to follow this with some more headers yourself, use the **-noel** (NO-Empty-Line) flag as shown in the next example.

- If you do not want the blank line after the header line, use the **-noel** flag:

```
cgiutils -noel -expires "2 days" -nodate
HTTP/1.0 200 Virtual doc follows
MIME-Version: 1.0
Server: IBM-ICS
Expires: Tuesday, 07-Jan-96 03:43:46 GMT
```

htadm command

Note: When you use `htadm` to add a user, change a password, or verify a password, you must enter the password on the command line. Because `htadm` destroys the password from the command line as soon as possible, it is highly unlikely that you can see a user's password by looking at the process listing on the machine (with the `psstat` command, for example).

Syntax

```
htadm -Flag [Modifier]
```

Flags

-adduser *password-file* *user-name* [*password* [*real-name*]]

Adds a user and password into the password file. If you enter the command with only *password-file*, you are prompted for the other parameters.

password-file

The path and name of the password file to which you want to add the user.

user-name

The name of the user you want to add.

Use only alphabetic and numeric characters for the user name; do not use special characters.

The command fails if there is already a user of the same name in the password file.

Commands

password

The password you want to define for the user name.

Passwords can be up to 32 characters long. Use only alphabetic and numeric characters for the password; do not use special characters.

Note: Some browsers are unable to read and send passwords longer than eight characters. Because of this limitation, if you define a password longer than eight characters, the server recognizes either the complete password or just the first eight characters of the password as valid.

real-name

A comment or name you want to use to identify the user name you are adding. Whatever you enter will be written into the password file.

-deluser *password-file* [*user-name*]

Deletes a user from the password file. If you enter the command with only *password-file*, you are prompted for the *user-name* parameter.

password-file

The path and name of the password file from which you want to delete a user.

user-name

The name of the user you want to delete. The command fails if the user name you specify does not exist in the password file.

-passwd *password-file* [*user-name* [*password*]]

Changes the password for a user name already defined in the password file. If you enter the command with only *password-file*, you are prompted for the other parameters.

password-file

The path and name of the password file that contains the user name whose password you want to change.

user-name

The user name whose password you want to change. The command fails if the user name you specify does not exist in the password file.

password

The new password you want to define for the user name.

Passwords can be up to 32 characters long. Use only alphabetic and numeric characters for the password; do not use special characters.

Note: Some browsers are unable to read and send passwords longer than eight characters. Because of this limitation, if you define a password longer than eight characters, the server recognizes either the complete password or just the first eight characters of the password as valid.

-check *password-file* [*user-name* [*password*]]

Verifies the password for a user name already defined in the password file and lets you know if it is correct or not. If you enter the command with only *password-file* you are prompted for the other parameters.

Commands

password-file

The path and name of the password file that contains the user name whose password you want to verify.

user-name

The user name whose password you want to verify. The command fails if the user name you specify does not exist in the password file.

password

The password that you want to verify. If the password you enter is the one defined for the user name, the command writes Correct to standard output and completes with a 0 return code. If the password you enter is not the one defined for the user name, the command writes Incorrect to standard output.

-create *password-file*

Create an empty password file.

password-file

The path and name of the password file that you want to create.

Examples

- To add a user to a password file:

```
htadm -adduser d:\tcip\etc\heroes.pwd clark superman "Clark Kent"
```

- To delete a user from a password file:

```
htadm -deluser d:\tcip\etc\heroes.pwd clark
```

htimage command

Use the htimage command to process clickable image maps. It allows defined regions within an image map to be associated with specific URLs. When users click on a defined region, the htimage command executes a CGI script that displays the URL associated with the region. When the server returns the URL associated with the region, it issues a 302 return code (Moved Temporarily) and a location header containing the URL.

Note: The URL specified in the location header must be fully qualified, or absolute. Include the protocol, host, and request path in the URL. For example:

```
http://hostname/filename.html
```

The Internet Connection Secure Server assumes, if the URL in the map file is not absolute, that the URL is local, and serves it directly. While this saves

Commands

some network overhead, the browser thinks the original request was satisfied and continues to use the original request as the base for future requests.

When the URL is not fully qualified and the server finds the file locally, the Internet Connection Secure Server issues a Content-Location header which some browsers use to make future requests.

Many browsers understand the HTML tag, `<BASE HREF= >`, which can be used to specify the base URL, (for example, `http://hostnamebase URL`). When this tag is used, URLs that are not fully qualified are evaluated relative to this base URL and are correctly found.

The command is used in conjunction with a map file. Map files are text files that define regions within a graphics file by their x,y coordinates and map them to the various URLs. For additional information on using `htimage` and creating map files and images, see the Web design information on the Internet Connection Family Web site (<http://www.ics.raleigh.ibm.com>). The Web design information is accessible from the Front Page of your server.

Note: Currently the universal image file accepted by all browsers and servers is the GIF format. GIF is an 8-bit 256 color image file.

You cannot use `htimage` from the command line. You include it as part of an anchor tag within an HTML document, and it is called when the server processes that document.

The `htimage` command is located in the **CGI Bin scripts directory** you specified at installation. The installation default is `C:\WWW\CGI-BIN`.

Syntax

Since `htimage` can only be called from an anchor tag within an HTML document, the syntax is shown as HTML markup.

```
<A HREF="/CGI-BIN/htimage/mapfile.txt">  
<IMG SRC="/ICONS/image.gif" ISMAP></A>
```

The `HREF` attribute of the anchor (`A`) tag specifies the URL of the `htimage` command followed by the URL of the map file.

This syntax description assumes that the default configuration file is being used. The default configuration file contains an `Exec` directive that maps requests beginning with `/CGI-BIN/` to the directory that contains the `htimage` command.

The server uses everything following `/htimage/` as the URL of the map file. If the server is using the default configuration and the URL contains only a file name, the server would look for the file in the document root directory.

Commands

The SRC attribute of the image (IMG) tag specifies the URL of the file that contains the graphic you want to use as an image map.

The ISMAP attribute indicates that the graphic is an image map.

Examples

Following is a description of the map files that must be used with the htmimage command. A map file is a text file. Each line of the file is in the following format:

region-identifier [*region-definition*] *URL*

region-identifier

A keyword that identifies the type of region being defined. Valid values are:

- rectangle
- circle
- polygon
- default

region-definition

A set of numbers that defines a particular region of the graphic. The format of the region definition is different for each type of region.

Coordinates within parentheses identify a point relative to the top left corner of the image. The first number is the number of pixels to the right of the top left corner. The second number is the number of pixels down from the top left corner. There are several shareware programs available that can help you easily identify the coordinates of particular points within a graphic file.

```
default URL
rectangle (x1,y1) (x2,y2) URL
circle (x,y) r URL
polygon (x1,y1) (x2,y2) (x3,y3) URL
```

- The default keyword does not define a region. The keyword is followed by a URL to link to when the client clicks on a portion of the image map that is not covered by any of the other region definitions.
- For rectangle, the first point is the upper left corner of the rectangle. The second point is the lower right corner. In other words, define any two diagonally opposite corners having coordinates (x1,y1) and (x2,y2).
- For circle, the point is the center of the circle. The single number following the point is the radius of the circle as measured in pixels.
- For polygon, up to 100 points can be defined. The shape is formed by connecting the points in the order they are given. The last point is connected to the first.

For example, you might use the following HTML in a document:

Commands

```
<A HREF="/cgi-bin/htimage/mapit.txt">  
<IMG SRC="/icons/mapimage.gif" ISMAP></A>
```

The above example calls the htimage command with a map file named mapit.txt. The mapit.txt file would define regions of the mapimage.gif graphic file. Because no path is specified for mapit.txt, the server would look for it on the document root directory. Following is an example of what mapit.txt might look like.

```
default          http://brimstone/cgi-bin/go_home  
rectangle (50, 100) (200,200)    http://brimstone/cgi-bin/go_to_it  
circle (100,300) 50              http://brimstone/pub/example.html  
polygon (450,350) (450,500) (150, 500) http://brimstone/pub/triangle.html
```

One example of a shareware program for determining the x,y coordinates is mapedit. You can obtain mapedit from the following URL:

<http://sunsite.unc.edu/boutell/mapedit/mapedit.html>

Please note that output from mapedit is in NCSA format. It is different from the CERN examples shown.

The same examples in NCSA format require anchor tags within an HTML document as follows:

```
<a href="/cgi-bin/imagemap/mapit.txt">  
</a>
```

The map file will be an EBCDIC text file with the following format:

```
default http://brimstone/cgi-bin/go_home  
rect    http://brimstone/cgi-bin/go_to_it    50, 100  200,200  
circle  http://brimstone/pub/example.html    100,300  100,350  
poly    http://brimstone/pub/triangle.html   450,350  450,500  150, 500
```

URL

The fully qualified or absolute URL, including the protocol, hostname and filename, is required. If the URL in the map file is not absolute, the Internet Connection Secure Server assumes that the URL is local and serves it directly.

httpd command

Use the httpd command to start the server. The default setting is to export the /Public directory tree.

You can set all these flags (except -r) using the directives in the server configuration file.

You can set the directory-browsing flags with the DirAccess configuration directive.

Commands

It is common practice to create a file named README containing instructions or notices to be read by anyone new to the directory. By default, httpd will embed any README file in the hypertext version of a directory. The README file instructions can also be set with the DirReadme configuration directive.

Syntax

```
httpd [-Flag [-Flag [-Flag..]]]
```

-B *bounce*

The server normally does not bind to its listen ports with the SO_REUSEADDR socket option. This helps to prevent running multiple instances of the server with the same Pid, log, and proxy cache files. When a server is shutdown or terminates abnormally, there may be sockets remaining in TIMED_WAIT state in the TCP/IP stack. The Internet Connection Secure Server retries the bind to its listen ports for up to two minutes to allow previously used sockets to close.

If you know that the prior instance of the server has terminated, you can use the -B flag to set SO_REUSEADDR on to the servers listen ports before binding to them. This avoids the TIMED_WAIT delay. This flag is especially useful if you have automation software that recognizes Internet Connection Secure Server termination and needs to start a replacement server as quickly as possible.

-db

Same as the **-dt** flag but puts the README at the bottom, after the listing. You can combine **-db** with the **-dy** and **-dt** options (for example, **-dyb**, **-dtb**).

-dn

Disables directory browsing. The server responds with an error message to requests for a directory listing.

-dr Disables the README inclusion feature.

-ds

Enables selective directory browsing. Directory browsing is allowed only for directories containing a file named .www_browsable.

-dt For any browsable directory that contains a README file, includes the text of the README file at the top of the document before the listing. This is the default setting. You can combine **-dt** with the **-db** and **-dy** options (for example, **-dtb**, **-dty**).

-dy

Enables directory browsing. By default, directories can be returned as hypertext documents. This is the default setting. You can combine -dy with the **-db** and **-dt** options (for example, **-dyb**, **-dty**).

-gc_only

Only does garbage collection and then exits. This flag is used only for caching proxy servers.

Commands

-l *log-file*

Specifies the file to use to log requests.

-nobg

Runs the server as a background process. Do not run the server as a background process.

-nosec

Specifies the flag to use to force a base server. Security loads are bypassed.

-nosnmp

Turns SNMP support off.

-p *port-number*

Listens on this port number. The default port number is 80. This flag overrides the Port directive specified in the configuration file.

-r *configuration-file*

Specifies the file to use as the configuration file. You must use this flag if you want to start the server with a configuration file other than the default configuration file.

Note: If you specify the -r option and either the normalmode or sslmode option, you **must** specify normalmode or sslmode before -r.

-snmp

Turns SNMP support on.

-sslmode [on|off]

For a secure server, turns on the SSL protocol.

-sslport [port]

For a secure server, sets the port used for the SSL protocol.

-version

Returns the version number of httpd and libwww (the WWW Common Library).

SIGTERM

This causes httpd to stop and exit when complete. You can use SIGKILL or CANCEL to immediately terminate.

SIGHUP

Restart a running httpd. This causes httpd to stop accepting new requests, complete current requests, and, if there are no errors, reload the configuration file and resume processing. If there are errors, you must fix the configuration file.

Examples

- To start the server on port 80, using the D:\TCP\IP\etc\httpd.config configuration file instead of the default configuration file:

```
httpd -r D:\TCP\IP\etc\httpd.config -p 80
```

If the Port directive is given in the configuration file, the **-p** flag is not required. The **-p** flag can be used to override the value set in the configuration file.

Appendix B. Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make them 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 that IBM product, program or service may be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594, U.S.A.

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:

Site Counsel
IBM Corporation
P.O. Box 12195
3039 Cornwallis
Research Triangle Park, NC 27709-2195
USA

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement.

This document is not intended for production use and is furnished as is without any warranty of any kind, and all warranties are hereby disclaimed including the warranties of merchantability and fitness for a particular purpose.

This product includes computer software created and made available by CERN. This acknowledgement shall be mentioned in full in any product which includes the CERN computer software included herein or parts thereof.

Notices

Programming interface information

This documentation is intended to help the Webmaster configure and operate the Internet Connection Secure Server (ICSS). This information documents General-use Programming Interface and Associated Guidance Information provided by ICSS.

General-use programming interfaces allow the user to use the HTTP and SSL protocols of ICSS.

Trademarks

The following terms are trademarks of IBM Corporation in the United States or other countries or both.

AIX

IBM

IBMLink

InfoExplorer

MVS/ESA

NetView

OpenEdition

OS/2

OS/390

OS/400

PS/2

RACF

TrackPoint

VTAM

WebExplorer

HP is a trademark of Hewlett-Packard Company.

Microsoft, Windows, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

Solaris is a trademark of Sun Microsystems, Inc.

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, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

Glossary

This glossary defines terms used by the Internet Connection Secure Server.

accessory script. A CGI script program that processes SEARCH, POST, PUT, or DELETE requests. The accessory scripts process requests that are not explicitly mapped to a CGI script program named on an EXEC directive.

address. The unique code assigned to each device or workstation connected to a network. A standard Internet address (or IP address) is a 32-bit address field. This field contains two parts. The first part is the network address; the second part is the host number. See also *IP address*.

agent. (1) In systems management, a user that, for a particular interaction, has assumed an agent role.
(2) An entity that represents one or more managed objects by (a) emitting notifications regarding the objects and (b) handling requests from managers for management operations to modify or query the objects.

alias. A name assigned to a server. The alias makes the server independent of the name of its host machine. The alias must be defined in the domain name server.

asymmetric keys. In secure communications, the two keys in a key pair. The keys are called asymmetric because one key holds more of the encryption pattern than the other does. See *key pair*.

authentication. In secure communications, a means of verifying the identity of a server or browser (client) with whom you wish to communicate. A sender's authenticity is demonstrated by the digital certificate issued to the sender. See also *certificate*.

browser. A client program that initiates requests to a server and displays the returned information.

cache. A special-purpose buffer storage, smaller and faster than main storage, used to hold a copy of data that may be frequently accessed. Use of a cache reduces access time, but may increase memory requirements.

caching proxy server. A proxy server that can store the documents it retrieves from other servers in a local cache. The server can then respond to subsequent

requests for the same documents without having to retrieve them from other servers. This can improve response time.

CERN. Conseil Europeen pour la Recherche Nucleaire (European Laboratory for Particle Physics). Located in Geneva, CERN initiated the World Wide Web, and was the first to create a Web server. The CERN Web server is the basis for many commercially available servers, such as IBM's Internet Connection Secure Server.

certificate. In secure communications, a digital document that binds an encryption key to the identity of the certificate owner, so that the certificate owner can be authenticated. A certificate is issued by a certification authority (CA). See also *encryption*, *certificate*, and *certification authority (CA)*.

certification authority (CA). In secure communications, a trusted third party (such as VeriSign, Inc.) or a designated internal authority who issues certificates. See also *certificate*.

CGI (common gateway interface). A standard interface between Web servers and external programs. CGI scripts are programs that use this interface to perform tasks not usually done by the server, such as form processing.

CGI program. A program that uses the common gateway interface (CGI) to perform tasks not usually done by the server, such as form processing. CGI programs can be written in any language supported by the operating system on which the server is run. The language can be a scripting language or a programming language.

client. A computer system or process that requests a service of another computer system or process. For example, a workstation or personal computer requesting HTML documents from an IBM Internet Connection Secure Server is a client of the IBM Internet Connection Secure Server it connects to.

configuration file. A file that describes the devices, optional features, communications parameters, and programs installed on a workstation. For Internet Connection Secure Server, the configuration file is named `httpd.cnf` and by default is put in the path

specified on the SET ETC statement in your CONFIG.SYS file. The configuration file contains directives that define the various settings for the server.

cookie. A general mechanism that server-side connections, such as CGI scripts, can use to store information on the client side of the connection for later retrieval. For example, a retail Web site can store per-user preferences on the client, and have the client supply those preferences every time that site is connected to. A cookie is introduced to the client in a Set-Cookie header, which is included as part of an HTTP response.

default. A value, attribute, or option that is assumed when none is explicitly specified.

directive. A statement used in the Internet Connection Secure Server configuration file to define a particular setting for the server.

directory. A named grouping of files in a file system.

Distinguished Name. In secure communications, the name and address of the person and organization to whom a certificate has been issued. See also *certificate*.

document root directory. The primary directory where a Web server stores accessible documents. When the server receives requests that do not point to a specific directory, it tries to serve the request from this directory.

domain. In an internet, a part of the naming hierarchy. A domain name consists of a sequence of names (labels) separated by periods (dots).

domain name. A name of a host system in a network. A domain name consists of a sequence of names (labels) separated by periods (dots).

domain name server. A server program that supplies address-to-name translation by mapping Internet addresses to domain names. Use of a domain name server allows users to request services of another computer using a symbolic name, which is easier to remember than an Internet address.

dotted-decimal notation. The syntactical representation for a 32-bit integer that consists of four 8-bit numbers, written in base 10 and separated by periods (dots). It is used to represent IP addresses.

dynamic link library (DLL). A file containing executable code and data bound to a program at load

time or run time. The code and data in a dynamic link library can be shared by several applications simultaneously.

encryption. In secure communications, a means of scrambling data to prevent the data from being read by anyone other than the intended recipient. The sender uses a key to encrypt the message; the recipient uses the decryption key. See also *key* and *key pair*.

file extension. The last part of a file's name, following the period (dot). For example, in the filename *welcome.html*, the file extension is *html*.

firewall. A computer that connects a private network, such as a business, to a public network, such as the Internet. It contains programs that limit the access between two networks. See also *proxy gateway*.

FTP (File Transfer Protocol). An application protocol used for transferring files to and from host computers. FTP requires a user ID, and a password to allow access to files on a remote host system.

gateway. A functional unit that connects a local data network with another network. See also *proxy gateway*.

Gopher. The protocol, developed at the University of Minnesota, that provides a menu-driven interface for accessing files and information on other computers.

home page. The welcome page on the document root directory of a Web server. Commonly used as the entry point for the contents of the server. See also *welcome page*.

host. A computer, connected to a network, which provides an access point to that network. A host can be a client, a server, or a client and server simultaneously.

host name. A name, such as *tcpipidd.raleigh.ibm.com*, that is defined for an IP address, such as *9.67.97.103*.

HTML (Hypertext Markup Language). A language used to create hypertext documents. Hypertext documents can include links to other related documents. HTML controls the format of text and position of form input areas, for example, as well as the navigable links.

HTML document. A document written in HTML that may contain links to other documents that contain additional information about related terms or subjects.

HTTP (Hypertext Transfer Protocol). The protocol used to transfer and display hypertext documents.

HTTP method. An action used by the Hypertext Transfer Protocol. For example, HTTP methods include GET, POST, and PUT.

icon. A graphical representation of an object (a file or program), consisting of an image, image background, and a label.

Internet. A wide area network connecting thousands of disparate networks in industry, education, government, and research. The Internet network uses TCP/IP as the standard for transmitting information.

IP address. The unique 32-bit address that specifies the actual location of each device or workstation in the Internet. For example, 9.67.97.103 is an IP address.

key. In secure communications, an algorithmic pattern used by a sender to encrypt messages, and by a recipient to decrypt messages. See also *encryption*, *key pair*, and *key ring*.

key pair. In secure communications, a public key and a private key. The sender uses the private key to encrypt the message; the recipient uses the public key to decrypt the message. Because the private key holds more of the encryption pattern than the public key does, the key pair is called asymmetric. See also *public key* and *private key*.

key ring. In secure communications, a file that contains public keys, private keys, trusted roots, and certificates. See also *public key*, *private key*, *trusted root*, and *certificate*.

managed node. In Internet communications, a workstation, server, or router that contains a network management agent. In the Internet Protocol (IP), the managed node usually contains a Simple Network Management Protocol (SNMP) agent.

method. An action used by the Hypertext Transfer Protocol. For example, HTTP methods include GET, POST, and PUT.

MIB. (1) Management Information Base. A collection of objects that can be accessed by means of a network management protocol. (2) A definition for management information that specifies the information available from a host or gateway and the operations allowed.

MIME (Multipurpose Internet Mail Extensions). An Internet standard for multimedia e-mail, including graphics, audio, and fax.

name server. A host that provides name resolution for a network. Name servers translate symbolic names assigned to networks and hosts into the Internet (IP) addresses used by machines.

network management station. In the Simple Network Management Protocol (SNMP), a station that executes management application programs that monitor and control network elements.

password. In secure communications, a string of characters that you use to protect access to your key ring. See also *key ring*.

path. A statement that indicates where a file is stored on a particular drive. The path consists of all the directories that must be opened to get to a particular file. The directory names are separated by the backslash (\).

persistent connection. A TCP/IP connection that allows the server to accept multiple requests and to send responses over the same connection.

PICS. Platform for Internet Content Selection. An evolving set of specifications governing the creation and use of ratings for Web information, including HTML files, as well as image, sound, and animation files. Content providers can rate and label their own Web information; also, independent rating services can rate Web information. Internet users can then request the ratings as a way to preview and filter Web information for acceptable content.

port. An end point for communication between applications, generally referring to a logical connection. A port provides queues for sending and receiving data. Each port has a port number for identification. When the port number is combined with an Internet address, it is called a socket address.

private key. In secure communications, an algorithmic pattern used to encrypt messages that can be decrypted only by the corresponding public key. A private key is also used to decrypt messages that were encrypted by the corresponding public key. You keep your private key on your own system in a key ring, protected by a password. See also *encryption*, *public key*, and *key ring*.

protection setup. A group of protection subdirectives that work together to define how the server should control access to the resources being protected. You can define protection setups within the configuration file, in separate protection setup files, or by using the Configuration and Administration forms.

protocol. The set of rules governing the operation of functional units of a communication system if communication is to take place. Protocols can determine low-level details of machine-to-machine interfaces, such as the order in which bits from a byte are sent; they can also determine high-level exchanges between application programs, such as file transfer.

proxy gateway. A type of firewall that protects computers in a business network from access by users outside that network. See also *firewall*.

proxy server. A server that can retrieve documents from other servers for its clients.

public key. In secure communications, 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. You broadcast your public key to everyone who will need to exchange encrypted messages with you. See also *encryption*, *private key*, and *key ring*.

README file. A file containing information and instructions for using the associated program or programs

request. The part of a URL that follows the protocol and server host name. For example, in the URL `http://www.server.com/rfoul/sched.html`, the request is: `/rfoul/sched.html`

server. A computer that provides shared services to other computers over a network; for example, a file server, a print server, or a mail server.

server root directory. The directory where the Internet Connection Secure Server program is installed. By default, the server root directory is `C:\WWW\BIN`.

server-side includes (SSI). A facility for including dynamic information in documents sent to clients, such as current date, the file's last modification date, and the size or last modification of other files.

SSL. Secure Sockets Layer. SSL is a popular security scheme developed by Netscape Communications Corp., along with RSA Data Security, Inc. SSL allows the client to authenticate the server and all data and requests to be encrypted. The URL of a secure server protected by

SSL begins with `https` (rather than `http`). See also *authentication*.

subdirectory. A directory contained within another directory in a file system hierarchy.

thread. The smallest unit of operation performed within a process.

thread pool. The threads that are either being used or available to Internet Connection Secure Server.

trusted root. In secure communications, the public key and associated Distinguished Name of a certification authority (CA). See also *public key*, *Distinguished Name*, and *certification authority (CA)*.

URL (Uniform Resource Locator). The address convention that indicates the location of an item on the World Wide Web. It includes the protocol followed by the fully-qualified host name, and the request. The server typically maps the request portion of the URL to a path and file name. For example, `http://www.ibm.com/index.html`

virtual host. One of several host names that you can define for a single IP address in the domain name server. That IP address can then serve multiple files, rather than requiring different IP addresses for different files.

WAIS (Wide Area Information Service). A network information system that enables clients to search documents on the World Wide Web.

Web. The World Wide Web: the network of HTTP servers that contain programs and files, such as hypertext documents that contain links to other documents on HTTP servers.

Web server. A server on the World Wide Web. See also *Web*.

welcome page. A document that is returned by a Web server in response to a request that points to a directory but does not contain a file name. Each accessible directory on the server can have a welcome page. See also *home page*.

wildcard character. An asterisk (*) used in a template. For the template to be matched, an asterisk can be replaced by any character string or single character. A question mark must be replaced by one single character.

Bibliography

This bibliography lists the books related to the Internet Connection Secure Server Version 4.2.

For AIX

- *Quick Beginnings*, GC31-8482
Explains how to plan for, install, start your server using the default configuration settings, and stop your server. Also explains how to view online help and print online books.
- *Webmaster's Guide for AIX, HP-UX, and Solaris*, GC31-8487
Explains how to change the default configuration settings to meet your needs, either by using the built-in configuration utility or by editing the configuration file. Also explains how to control and track users' access to your server, how to include dynamic information in the files your server returns to users, and how to set up a secure environment for your users to conduct business.
- *Web Programming Guide* (available in HTML format from your server's Front Page or in PDF format from this URL: <http://www.ics.raleigh.ibm.com>)
Explains how to write external programs that interact with the Internet Connection Secure Server, via either the Common Gateway Interface (CGI) or the Internet Connection API (ICAPI). For example, you can write a CGI program to generate a dynamic response to user input, or you can write an ICAPI program to customize the way errors are handled.

For HP-UX

- *Quick Beginnings*, GC31-8483
Explains how to plan for, install, start your server using the default configuration settings, and stop your server. Also explains how to view online help and print online books.
- *Webmaster's Guide for AIX, HP-UX, and Solaris*, GC31-8487
Explains how to change the default configuration settings to meet your needs, either by using the built-in configuration utility or by editing the configuration file. Also explains how to control and track users' access to your server, how to include dynamic information in the files your server returns to users, and how to set up a secure environment for your users to conduct business.
- *Web Programming Guide* (available in HTML format from your server's Front Page or in PDF format from this URL: <http://www.ics.raleigh.ibm.com>)
Explains how to write external programs that interact with the Internet Connection Secure Server, via either the Common Gateway Interface (CGI) or the Internet Connection API (ICAPI). For example, you can write a CGI program to generate a dynamic response to user input, or you can write an ICAPI program to customize the way errors are handled.

For OS/2 Warp

- *Quick Beginnings*, GC31-8481
Explains how to plan for, install, start your server using the default configuration settings, and stop your server. Also explains how to view online help and print online books.
- *Webmaster's Guide*, GC31-8486

Explains how to change the default configuration settings to meet your needs, either by using the built-in configuration utility or by editing the configuration file. Also explains how to control and track users' access to your server, how to include dynamic information in the files your server returns to users, and how to set up a secure environment for your users to conduct business.

- *Web Programming Guide* (available in HTML format from your server's Front Page or in PDF format from this URL: <http://www.ics.raleigh.ibm.com>)

Explains how to write external programs that interact with the Internet Connection Secure Server, via either the Common Gateway Interface (CGI) or the Internet Connection API (ICAPI). For example, you can write a CGI program to generate a dynamic response to user input, or you can write an ICAPI program to customize the way errors are handled.

- *SystemView for OS/2 Version 1.1 Up and Running!*, SH19-4184

Explains the concepts of systems management in a LAN workgroup environment. Also explains how to install and configure SystemView for OS/2, and includes user scenarios to show you how to get started with day-to-day systems management and software distribution.

dynamic information in the files your server returns to users, and how to set up a secure environment for your users to conduct business.

- *Web Programming Guide* (available in HTML format from your server's Front Page or in PDF format from this URL: <http://www.ics.raleigh.ibm.com>)

Explains how to write external programs that interact with the Internet Connection Secure Server, via either the Common Gateway Interface (CGI) or the Internet Connection API (ICAPI). For example, you can write a CGI program to generate a dynamic response to user input, or you can write an ICAPI program to customize the way errors are handled.

Explains how to plan for, install, start your server using the default configuration settings, and stop your server. Also explains how to view online help and print online books.

Explains how to change the default configuration settings to meet your needs, either by using the built-in configuration utility or by editing the configuration file. Also explains how to control and track users' access to your server, how to include dynamic information in the files your server returns to users, and how to set up a secure environment for your users to conduct business.

- *Web Programming Guide* (available in HTML format from your server's Front Page or in PDF format from this URL: <http://www.ics.raleigh.ibm.com>)

Explains how to write external programs that interact with the Internet Connection Secure Server, via either the Common Gateway Interface (CGI) or the Internet Connection API (ICAPI). For example, you can write a CGI program to generate a dynamic response to user input, or you can write an ICAPI program to customize the way errors are handled.

For Solaris

- *Quick Beginnings*, GC31-8484

Explains how to plan for, install, start your server using the default configuration settings, and stop your server. Also explains how to view online help and print online books.

- *Webmaster's Guide for AIX, HP-UX, and Solaris*, GC31-8487

Explains how to change the default configuration settings to meet your needs, either by using the built-in configuration utility or by editing the configuration file. Also explains how to control and track users' access to your server, how to include

Related publications

- *OS/390 MVS System Management Facilities*, GC28-1783

Index

A

- access control directives
 - DefProt 77
 - Protect 79
 - Protection 83
 - Protection subdirectives
 - ACLOverride 84
 - AuthType 85
 - DeleteMask 85
 - GetMask 85
 - GroupFile 85
 - Mask 86
 - PasswdFile 86
 - PostMask 86
 - PutMask 87
 - ServerID 87
- access control lists (ACL) files
 - limiting access to individual files 214
 - server group files 214
 - using 215
- access log
 - maintenance options 156
 - path 155
 - Server Activity Monitor 236
 - set filters 157
- AccessLog directive 54
- AccessLogArchive directive 55
- AccessLogExcludeMethod directive 57
- AccessLogExcludeMimeType directive 57
- AccessLogExcludeReturnCode directive 58
- AccessLogExcludeURL directive 56
- AccessLogExpire directive 58
- AccessLogSizeLimit directive 59
- AccessReportDescription directive 59
- AccessReportExcludeHostName directive 61
- AccessReportExcludeMethod directive 62
- AccessReportExcludeReturnCode directive 62
- AccessReportExcludeURL directive 60
- AccessReportIncludeHostName directive 61
- AccessReportIncludeURL directive 60
- AccessReportRoot 63
- AccessReportRoot directive 63
- AccessReportTemplate directive 63
- AccessReportTopList directive 64
- accountability 241
- ACLOverride subdirective 84, 214
- activity statistics
 - network 235
 - server 235
- AddBlankIcon directive 41
- AddCharSet directive 94
- AddClient directive 99
- AddDirIcon directive 42
- AddEncoding directive 94
- AddIcon directive 43
- AddLanguage directive 93
- AddParentIcon directive 43
- address template protection 205
- Address, Multiple IP 19
- AddType directive 95
- AddUnknownIcon directive 44
- administration forms 4
- AgentLog directive 55, 64
- AlwaysWelcome directive 45
- authentication 208, 242
- Authentication directive 124
- authenticity 241
- Authorization directive 125
- AuthType subdirective 85, 208
- autostart server 23

B

- backing up files 22
- badredirect 113
- badrequest 112
- badscript 112
- baduser 113
- basic authentication 208
- basic directives
 - BindSpecific 37
 - DNS-Lookup 38
 - HostName 38
 - imbeds 39
 - Port 40
 - ServerRoot 41
- BindSpecific directive 37

byrule 113

C

CA

See Certification Authority (CA)

CacheAccessLog directive 65

CacheDefaultExpiry directory 134

CacheExpiryCheck directive 134

CacheLastModifiedFactor directive 135

CacheLimit_1 directive 135

CacheLimit_2 directive 136

CacheLocalFile directive 145

CacheLocalMaxBytes directive 145

CacheLocalMaxFiles directive 146

CacheLockTimeOut directive 136

CacheNoConnect directive 136

CacheOnly directive 137

CacheRoot directive 137

CacheSize directive 138

CacheUnused directive 138

Caching directive 138

caching proxy server

configuring

designating port number 14

other proxy settings 15

proxy server as 13

specifying protocols 13

controlling use 17

overview 11

central file management for PICS 192

certificates

digital 242

managing 245

certificates, digitally signed

buying certificate from a CA 251

issuing certificates yourself 252

certificates, obtaining 251

Certification Authority (CA)

designating key as Trusted Root 253, 274

procedure for becoming your own

creating a key pair 270

requesting a certificate 270

processing a certificate

overview 276

receiving server certificate 278

sending client and server processed

certificates 280

using certutil command 277

receiving a certificate 273

certutil command 277, 287

cgi_error log 66

CGIErrorLog directive 66

cgiparse command 288

cgutils command 291

command prompt, starting from 24

commands

certutil 277, 287

cgiparse 288

cgutils 291

htadm 293

htimage 295

httpd 298

commands and protocols, SNMP 224

Common Gateway Interface (CGI) programs

as a security exposure

on a Web server 202

encryption modes 284

community names, SNMP 150, 233

confidentiality 240

configuration file

access control 76

basic 37

directories and welcome page 41

error message customization 111

ICAPI application processing 122

logs and reports 54

meta-information 120

methods 118

performance settings 144

proxy server settings 134

resource mapping 101

security 89

timeouts 116

user directories 52

configuration file syntax 198

configuration forms 4

configuring

caching proxy server 13

other proxy settings 15

proxy server as caching proxy server 13

configuring the Internet Connection Secure Server

changing the document root directory 8

editing configuration file 5

running your server as a caching proxy 11

using Configuration and Administration Forms 4

CONNECT method 118

controlling access 6

counter, how to display on a Web page 171

- Create Key and Request Certificate form
 - choose a CA 258
 - create public-private key pair 259
 - process your CA form type request 259
 - process your completed CA request 265
 - process your request 266
 - save copy of certificate request 264
 - shut down and restart server 266
 - specify a key ring password 260
 - specify distinguished name (DN) 261
 - specify key name 259
 - specify key ring 259
 - specify mailing option 263
 - specify receive certificate information 266
 - using 270
- creating an e-mail address for SNMP problem reports 151, 232
- creating PICS labels 195
- creating your home page 9
- criteria for rating web sites 188
- customizing a Web site
 - displaying page count, date, and time 171
 - sending customized pages 99
 - using server-side includes 178

D

- DataFilter directive 128
- date and time, how to display on a Web page 171
- default error messages 112
- default list, welcome pages 9
- DefineLBSservice paragraph 197, 199
- DefineService paragraph 197, 199
- DefProt directive 77
- DELETE method 118
- DeleteMask subdirective 85, 210
- designate trusted root keys form 275
- designating a port number (proxy server) 14
- digital certificate 243
- digital signature (fingerprint) 243
- DirAccess directive 45
- dirbrowse 114
- directives
 - AccessLog 54
 - AccessLogArchive 55
 - AccessLogExcludeMethod 57
 - AccessLogExcludeMimeType 57
 - AccessLogExcludeReturnCode 58
 - AccessLogExcludeURL 56
 - AccessLogExpire 58

- directives (*continued*)
 - AccessLogSizeLimit 59
 - AccessReportDescription 59
 - AccessReportExcludeHostName 61
 - AccessReportExcludeMethod 62
 - AccessReportExcludeReturnCode 62
 - AccessReportExcludeURL 60
 - AccessReportIncludeHostName 61
 - AccessReportIncludeURL 60
 - AccessReportRoot 63
 - AccessReportTemplate 63
 - AccessReportTopList 64
 - AddBlankIcon 41
 - AddCharSet 94
 - AddClient 99
 - AddDirIcon 42
 - AddEncoding 94
 - AddIcon 43
 - AddLanguage 93
 - AddParentIcon 43
 - AddType 95
 - AddUnknownIcon 44
 - AgentLog 55, 64
 - AlwaysWelcome 45
 - Authentication 124
 - Authorization 125
 - BindSpecific 37
 - CacheAccessLog 65
 - CacheDefaultExpiry 134
 - CacheExpiryCheck 134
 - CacheLastModifiedFactor 135
 - CacheLimit_1 135
 - CacheLimit_2 136
 - CacheLocalFile 145
 - CacheLocalMaxBytes 145
 - CacheLocalMaxFiles 146
 - CacheLockTimeOut 136
 - CacheNoConnect 136
 - CacheOnly 137
 - CacheRoot 137
 - CacheSize 138
 - CacheUnused 138
 - Caching 138
 - CGIErrorLog 66
 - DataFilter 128
 - DefProt 77
 - DirAccess 45
 - DirReadme 46
 - DirShowBrackets 46
 - DirShowBytes 47

directives (*continued*)

- DirShowCase 47
- DirShowDate 47
- DirShowDescription 48
- DirShowHidden 48
- DirShowIcons 48
- DirShowMaxDescrLength 49
- DirShowMaxLength 49
- DirShowMinLength 49
- DirShowSize 50
- Disable 119
- DNS-Lookup 38
- Enable 119
- EnableJavaServletSupport 131
- Error 129
- ErrorLog 66
- ErrorLogArchive 67
- ErrorLogExpire 68
- ErrorLogSizeLimit 69
- ErrorMessage 111
- Exec 101, 203
- Fail 103
- ftp_proxy 139
- Gc 139
- GcDailyGc 139
- GcMemUsage 140
- gopher_proxy 140
- HomeDir 52
- HostName 38
- http_proxy 141
- IconPath 50
- imbeds 39
- InheritEnv 109
- InputTimeout 116
- KeyFile 89
- LiveLocalCache 146
- Log 128
- LogFormat 69
- LogTime 70
- LogToGUI 70
- Map 104
- MaxActiveJavaThreads 132
- MaxActiveThreads 147
- MaxContentLengthBuffer 141
- MaxPersistRequest 148
- MetaDir 121
- MetaSuffix 121
- NameTrans 124
- no_proxy 141
- NoCaching 142

directives (*continued*)

- NoLog 71
- NormalMode 89
- ObjectType 126
- OutputTimeout 117
- overview 36
- Pass 106, 203
- PersistTimeout 148
- PICSDBLookup 127
- Port 40
- PostExit 129
- PreExit 123
- Protect 79
- Protection 83
- Protection subdirectives
 - ACLOverride 84, 214
 - AuthType 85, 208
 - DeleteMask 85
 - GetMask 85
 - GroupFile 85, 209
 - Mask 86
 - PasswdFile 86, 209
 - PostMask 86
 - PutMask 87
 - ServerID 87, 208
- ProxyAccessLog 142
- Redirect 108
- RefererLog 55, 71
- ReportDataArchive 74
- ReportDataCompressionProgram 72
- ReportDataExpire 76
- ReportDataSizeLimit 74
- ReportDataUnCompressionProgram 72
- ReportDataUnCompressionSuffix 73
- ReportProcessOldLogs 73
- ScriptTimeout 117
- ServerInit 123
- ServerPriority 147
- ServerRoot 41
- ServerTerm 130
- Service 126
- Servlet 133
- ServletDir 133
- ServletLog 132
- SNMP 150
- SNMPCommunityName 150
- SocksServer 143
- SSLClientAuth 90
- SSLMode 90
- SSLPort 91

directives (*continued*)

- SuffixCaseSense 99
 - UseACLS 149
 - UseMetaFiles 149
 - UserDir 53
 - wais_proxy 143
 - WebMasterEmail 151
 - Welcome 50
- directory and welcome page directives
- AddBlankIcon 41
 - AddDirIcon 42
 - AddIcon 43
 - AddParentIcon 43
 - AddUnknownIcon 44
 - AlwaysWelcome 45
 - DirAccess 45
 - DirReadme 46
 - DirShowBrackets 46
 - DirShowBytes 47
 - DirShowCase 47
 - DirShowDate 47
 - DirShowDescription 48
 - DirShowHidden 48
 - DirShowIcons 48
 - DirShowMaxDescrLength 49
 - DirShowMaxLength 49
 - DirShowMinLength 49
 - DirShowSize 50
 - IconPath 50
 - UserDir 53
 - Welcome 50
- DirReadme directive 46
- DirShowBrackets directive 46
- DirShowBytes directive 47
- DirShowCase directive 47
- DirShowDate directive 47
- DirShowDescription directive 48
- DirShowHidden directive 48
- DirShowIcons directive 48
- DirShowMaxDescrLength directive 49
- DirShowMaxLength directive 49
- DirShowMinLength directive 49
- DirShowSize directive 50
- Disable 119
- Disable directive 119
- Distinguished Name (DN)
used in authentication 243
- DNS-Lookup directive 38
- document root directory
changing 8

document root directory (*continued*)

- creating your home page 9
- default list of welcome pages 9
- description 8
- domain name 38
- dotdot 114

E

- e-mail address for SNMP problem reports 151, 232
- editing configuration file 5
- editing configuration files 197
- Enable 119
- Enable directive 119
- EnableJavaServletSupport directive 131
- encryption
 - overview 241
 - requests messages 242
 - response messages 242
 - SSL modes 283
- error condition key words
 - badredirect 113
 - badrequest 112
 - badscript 112
 - baduser 113
 - byrule 113
 - defined 111
 - dirbrowse 114
 - dotdot 114
 - ipmask 114
 - ipmaskproxy 114
 - methoddisabled 114
 - multifail 115
 - noacl 114
 - noentry 114
 - noformat 116
 - notallowed 114
 - notauthorized 113
 - notmember 113
 - okredirect 112
 - openfailed 115
 - proxyfail 113
 - scriptio 115
 - scriptnotfound 116
 - scriptstart 116
 - setuperror 115
 - unknownmethod 113
- Error directive 129
- error log
 - maintenance options 159

- error log (*continued*)
 - path 159
 - specify path 159
- error messages
 - conditions, causes, and default messages 112
 - customization 111
 - ErrorPage directive 111
 - key word
 - badredirect 113
 - badrequest 112
 - badscript 112
 - baduser 113
 - byrule 113
 - defined 111
 - dirbrowse 114
 - dotdot 114
 - ipmask 114
 - ipmaskproxy 114
 - methoddisabled 114
 - multifail 115
 - noacl 114
 - noentry 114
 - noformat 116
 - notallowed 114
 - notauthorized 113
 - notmember 113
 - okredirect 112
 - openfailed 115
 - proxyfail 113
 - scriptio 115
 - scriptnotfound 116
 - scriptstart 116
 - setuperror 115
 - unknownmethod 113
 - overview 111
- ErrorLog directive 66
- ErrorLogArchive directive 67
- ErrorLogExpire directive 68
- ErrorLogSizeLimit directive 69
- ErrorPage directive 111
- Exec directive 101, 203
- export versions, differences with US-Canadian 283
- exporting a key 245

F

- Fail directive 103
- files, backing up 22
- filters for access log
 - defaults 158

- filters for access log (*continued*)
 - external hits information 157
 - overview 157
 - reduce log size 157
 - Website access information 158
- finger 201
- firewall 11, 202
- forms, configuration and administration 4
- ftp_proxy directive 139

G

- Gc directive 139
- GcDailyGc directive 139
- GcMemUsage directive 140
- GET method 118
- GetMask subdirective 85, 210
- global settings for logs 155
- gopher_proxy directive 140
- group files 214
- GroupFile subdirective 85, 209

H

- HEAD method 118
- headers
 - MIME headers 120
 - no-parse header programs 291
- hints and tips
 - using the Server Activity Monitor 235
- home (welcome) page
 - creating 9
 - default list 9
 - default, changing 7
- HomeDir directive 52
- HostName directive 38
- how to rate a web site 194
- htadm command 209, 293
- htimage command 295
- HTTP headers 291
- HTTP methods
 - CONNECT 118
 - DELETE 118
 - Disable 119
 - Enable 119
 - GET 118
 - HEAD 118
 - OPTIONS 118
 - POST 118
 - PUT 118

HTTP methods (*continued*)
TRACE 119
http_proxy directive 141
httpd command 298

I

icon, starting from 24
IconPath directive 50
Imbeds directive 39
includes, server-side
format 179
preparing to use 178
using 178
InheritEnv directive 109
InputTimeout directive 116
integrity 240
Internet Connection Application Programming Interface
(ICAPI) directives
Authentication 124
Authorization 125
DataFilter 128
Error 129
Log 128
NameTrans 124
ObjectType 126
overview 122
PICSDBLookup 127
PostExit 129
PreExit 123
ServerInit 123
ServerTerm 130
Service 126
Internet Connection Secure Server 223
activity statistics 235
configuring
editing configuration file 5
using Configuration and Administration Forms 4
managing your 223
registering new servers 10, 11
starting
at initialization 23
automatically 23
from command prompt 24
from the icon 24
restarting 25
starting and stopping the server 23
starting multiple instances 24
starting and stopping 23
stopping 25

IP Address, Multiple 19
ipmask 114
ipmaskproxy 114

K

key management form 274
key pair
creating 270
designate trusted root keys form 274
key management form 270
managing 245
overview 270
key words, error condition
badredirect 113
badrequest 112
badscript 112
baduser 113
byrule 113
defined 111
dirbrowse 114
dotdot 114
ipmask 114
ipmaskproxy 114
methoddisabled 114
multifail 115
noacl 114
noentry 114
noformat 116
notallowed 114
notauthorized 113
notmember 113
okredirect 112
openfailed 115
proxyfail 113
scriptio 115
scriptnotfound 116
scriptstart 116
setuperror 115
unknownmethod 113
KeyFile directive 89
keys
asymmetric 242
deleting 249
designating a key as a trusted root 253
exporting 249
importing 250
key length 283
key pair 242
key ring 245

- keys (*continued*)
 - management 245, 248
 - password, key ring 246
 - private 242
 - public 242
 - removing a trusted root key 253
 - setting default key 248
 - symmetric 241
 - using 245

L

- LabelsFor paragraph 197, 198
- LiveLocalCache directive 146
- Log directive 128
- log maintenance options
 - access log 156
 - error log 159
- log paths, specifying
 - for access log 155
 - for error log 159
 - for proxy log 161
- LogFormat directive 69
- logging and reporting
 - logs
 - access log filters 157
 - access log maintenance 156
 - configuring scenario 160
 - error log maintenance 159
 - global settings 155
 - overview 153
 - path for access logs 155
 - path for error logs 159
 - path for proxy logs 161
 - reports
 - overview 161
 - sample scenarios 165
 - update templates 164
 - view default templates 165
- logs and reports directives
 - AccessLog 54
 - AccessLogArchive 55
 - AccessLogExcludeMethod 57
 - AccessLogExcludeMimeType 57
 - AccessLogExcludeReturnCode 58
 - AccessLogExcludeURL 56
 - AccessLogExpire 58
 - AccessLogSizeLimit 59
 - AccessReportDescription 59
 - AccessReportExcludeHostName 61

- logs and reports directives (*continued*)
 - AccessReportExcludeMethod 62
 - AccessReportExcludeReturnCode 62
 - AccessReportExcludeURL 60
 - AccessReportIncludeHostName 61
 - AccessReportIncludeURL 60
 - AccessReportRoot 63
 - AccessReportTemplate 63
 - AccessReportTopList 64
 - AgentLog 55, 64
 - CacheAccessLog 65
 - CGIErrorLog 66
 - Errorlog 66
 - ErrorLogArchive 67
 - ErrorLogExpire 68
 - ErrorLogSizeLimit 69
 - LogFormat 69
 - LogTime 70
 - LogToGUI 70
 - NoLog 71
 - RefererLog 55, 71
 - ReportDataArchive 74
 - ReportDataCompressionProgram 72
 - ReportDataExpire 76
 - ReportDataSizeLimit 74
 - ReportDataUnCompressionProgram 72
 - ReportDataUnCompressionSuffix 73
 - ReportProcessOldLogs 73
- LogTime directive 70
- LogToGUI 70

M

- management directives, system
 - SNMP 150
 - SNMPCommunityName 150
 - WebMasterEmail 151
- management information base, SNMP 224
- managing pics from central files 192
- managing PICS labels 189
- managing your keys, certificates, and trusted roots 245
- Map directive 104
- mapping, resource
 - Exec 101
 - Fail 103
 - InheritEnv 109
 - Map 104
 - Pass 106
 - Redirect 108

- Mask subdirective 86, 210
- MaxActiveJavaThreads directive 132
- MaxActiveThreads directive 147
- MaxContentLengthBuffer directive 141
- MaxPersistRequest directive 148
- messages, encryption
 - request 242
 - response 242
- messages, error
 - conditions, causes, and default messages 112
 - customization 111
 - ErrorPage directive 111
 - key word
 - badredirect 113
 - badrequest 112
 - badscript 112
 - baduser 113
 - byrule 113
 - defined 111
 - dirbrowse 114
 - dotdot 114
 - ipmask 114
 - ipmaskproxy 114
 - methoddisabled 114
 - multifail 115
 - noacl 114
 - noentry 114
 - noformat 116
 - notallowed 114
 - notauthorized 113
 - notmember 113
 - okredirect 112
 - openfailed 115
 - proxyfail 113
 - scriptio 115
 - scriptnotfound 116
 - scriptstart 116
 - setuperror 115
 - unknownmethod 113
 - overview 111
- meta-information directives
 - MetaDir 121
 - MetaSuffix 121
- MetaDir directive 121
- MetaSuffix directive 121
- methoddisabled 114
- methods, HTTP
 - CONNECT 118
 - DELETE 118
 - Disable 119

- methods, HTTP (*continued*)
 - Enable 119
 - GET 118
 - HEAD 118
 - OPTIONS 118
 - POST 118
 - PUT 118
 - TRACE 119
- MIB, SNMP 224
- monitor performance and status with Server Activity Monitor
 - access log 236
 - hints and tips 235
 - network activity statistics 235
 - server activity statistics 235
 - using 234
- multi-format processing directives
 - AddCharSet 94
 - AddClient 99
 - AddEncoding 94
 - AddLanguage 93
 - AddType 95
 - SuffixCaseSense 99
- multifail 115
- Multiple IP Address 19
- Multipurpose Internet Mail Extension (MIME)
 - headers 120

N

- NameTrans directive 124
- negative string 36
- network activity statistics 235
- no-parse header programs 291
- no_proxy directive 141
- noacl 114
- NoCaching directive 142
- noentry 114
- noformat 116
- NoLog directive 71
- NormalMode directive 89
- notallowed 114
- notauthorized 113
- notmember 113
- nph-programs 291, 293

O

- object IDs, SNMP MIB 224

- ObjectType directive 126
- obtaining a certificate 251
- okredirect 112
- online configuration and administration forms 196
- openfailed 115
- OPTIONS method 118
- OutputTimeout directive 117

P

- packet filtering 202
- Pass directive 203
- passing requests 203
- PasswdFile subdirective 86, 209
- password
 - community name, SNMP 150, 233
 - key ring
 - changing 246
 - specifying 260
 - stashing 247, 260
- password files 293
- performance setting directives
 - CacheLocalFile 145
 - CacheLocalMaxBytes 145
 - CacheLocalMaxFiles 146
 - LiveLocalCache 146
 - MaxActiveThreads 147
 - MaxPersistRequest 148
 - PersistTimeout 148
 - ServerPriority 147
 - UseACLs 149
 - UseMetaFiles 149
- PICS for rating services and label bureaus 191
- PICS for web site administrators 189
- PICSDBLookup directive 127
- Platform for Internet Content Selection (PICS) 187
 - central files 192
 - creating labels 195
 - DefineLBSERVICE paragraph 197, 199
 - DefineService paragraph 197, 199
 - definition 187
 - editing PICS configuration files 197
 - LabelsFor paragraph 197, 198
 - PICS label retrieval directive 127
 - rating files 195
 - starting a PICS service 194
 - storing on server 193
 - syntax 198
 - updating configuration file 196
 - using online configuration and administration forms 196
- Platform for Internet Content Selection (PICS)
 - (continued)*
 - using wildcards 200
- Port directive 40
- port number for proxy server 14
- positive string 36
- POST method 118
- Post, HTTP method 118
- PostExit directive 129
- PostMask subdirective 86, 210
- PreExit directive 123
- private keys
 - creating 270
 - designate trusted root keys form 274
 - key management form 270
 - managing 245
 - overview 270
- processing certificates as a CA 276
- Protect directive 79, 203
- protecting your server
 - access to individual files
 - ACL files, using 215
 - limiting 214
 - using server group files 214
 - activating protection 203
 - address template 205
 - creating protection setups
 - using config and admin forms 207
 - using config file 207
 - using separate setup files 207
 - examples 218
 - identifying setup to requesters 208
 - passing requests 203
 - pointing to a server group file 209
 - pointing to password file 209
 - processing requests 205
 - rules for user names, groups, and addresses 210, 211
 - specifying authentication type 208
 - specifying SSL client authentication parameters 212
 - specifying valid user names, groups, and addresses 210
 - types of 204
 - user name and password 204
- Protection directive 83
- Protection subdirectives
 - ACLOverride 84
 - AuthType 85
 - DeleteMask 85
 - GetMask 85

- Protection subdirectives (*continued*)
 - GroupFile 85
 - Mask 86
 - PasswdFile 86
 - PostMask 86
 - PutMask 87
 - ServerID 87
- protocols and commands, SNMP 224
- proxy
 - ftp_proxy 139
 - gopher_proxy 140
 - http_proxy 141
 - no_proxy 141
 - proxy server 11
 - wais_proxy 143
- proxy authentication 17
- proxy log path 161
- proxy server protection 202
- proxy server settings directives
 - CacheDefaultExpiry 134
 - CacheExpiryCheck 134
 - CacheLastModified Factor 135
 - CacheLimit1 135
 - CacheLimit2 136
 - CacheLockTimeOut 136
 - CacheNoConnect 136
 - CacheOnly 137
 - CacheRoot 137
 - CacheSize 138
 - CacheUnUsed 138
 - Caching 138
 - ftp_proxy 139
 - Gc 139
 - GcDailyGc 139
 - GcMemUsage 140
 - gopher_proxy 140
 - http_proxy 141
 - MaxContentLengthBuffer 141
 - no_proxy 141
 - NoCaching 142
 - ProxyAccessLog 142
 - SocksServer 143
 - wais_proxy 143
- ProxyAccessLog directive 142
- proxyfail 113
- public keys
 - creating 270
 - designate trusted root keys form 274
 - key management form 270
 - managing 245

- public keys (*continued*)
 - overview 270
- PUT
 - method 118
 - with server 118
- Put, HTTP method 118
- PutMask subdirective 87, 210

Q

- QUERY_STRING environment variable 288

R

- rating criteria 188
- rating file for PICS 192
- rating service for web site 187
 - how to rate web sites 187
- Rating web servers 187
- Redirect directive 108
- RefererLog directive 55, 71
- remote configuration and administration 4
- report filters
 - external hits information 163
 - most visted Web pages 164
 - reduce report scope 163
 - Website access information 163
- ReportDataArchive directive 74
- ReportDataCompressionProgram directive 72
- ReportDataExpire directive 76
- ReportDataSizeLimit directive 74
- ReportDataUnCompressionProgram directive 72
- ReportDataUnCompressionSuffix directive 73
- reporting and logging
 - logs
 - access log filters 157
 - access log maintenance 156
 - configuring scenario 160
 - error log maintenance 159
 - global settings 155
 - overview 153
 - path for access logs 155
 - path for error logs 159
 - path for proxy logs 161
 - reports
 - overview 161
 - sample scenarios 165
 - update templates 164
 - view default templates 165

- ReportProcessOldLogs directive 73
- reports
 - overview 161
 - sample scenarios 165
 - update templates 164
 - view default templates 165
- reports and logs directives
 - AccessLog 54
 - AccessLogArchive 55
 - AccessLogExcludeMethod 57
 - AccessLogExcludeMimeType 57
 - AccessLogExcludeReturnCode 58
 - AccessLogExcludeURL 56
 - AccessLogExpire 58
 - AccessLogSizeLimit 59
 - AccessReportDescription 59
 - AccessReportExcludeHostName 61
 - AccessReportExcludeMethod 62
 - AccessReportExcludeReturnCode 62
 - AccessReportExcludeURL 60
 - AccessReportIncludeHostName 61
 - AccessReportIncludeURL 60
 - AccessReportRoot 63
 - AccessReportTemplate 63
 - AccessReportTopList 64
 - AgentLog 55, 64
 - CacheAccessLog 65
 - CGIErrorLog 66
 - Errorlog 66
 - ErrorLogArchive 67
 - ErrorLogExpire 68
 - ErrorLogSizeLimit 69
 - LogFormat 69
 - LogTime 70
 - LogToGUI 70
 - NoLog 71
 - RefererLog 55, 71
 - ReportDataArchive 74
 - ReportDataCompressionProgram 72
 - ReportDataExpire 76
 - ReportDataSizeLimit 74
 - ReportDataUnCompressionProgram 72
 - ReportDataUnCompressionSuffix 73
 - ReportProcessOldLogs 73
- request processing, server 205
- resource mapping
 - Exec 101
 - Fail 103
 - InheritEnv 109
 - Map 104

- resource mapping (*continued*)
 - Pass 106
 - Redirect 108
- restarting from forms 25
- restarting server from window 25
- rlogin 201
- root directory for documents
 - changing 8
 - creating your home page 9
 - default list of welcome pages 9
 - description 8

S

- sample scenarios
 - accesses report (excluding beta and alpha7 requests) 167
 - configuring log files 160
 - department server accesses report (except internal addresses) 168
 - page hits report 165
 - PUT requests to beta subdirectory report 166
- scriptio 115
- scriptnotfound 116
- scriptstart 116
- ScriptTimeout directive 117
- Secure Sockets Layer
 - See* SSL (Secure Sockets Layer)
- security concepts
 - accountability 241
 - authenticity 241
 - confidentiality 240
 - integrity 240
 - the internet and security 240
 - what is security 240
- security directives
 - KeyFile 89
 - NormalMode 89
 - SSLClientAuth 90
 - SSLMode 90
 - SSLPort 91
- security function
 - using SSL client authentication
 - imbeds 39
 - in ACL files 85
 - KeyFile 89
 - NormalMode 89
 - SSLMode 90
 - SSLPort 91

- server 223
 - activity statistics 235
 - configuring
 - editing configuration file 5
 - using Configuration and Administration Forms 4
 - managing your 223
 - registering new servers 10, 11
 - starting
 - at initialization 23
 - automatically 23
 - from command prompt 24
 - from the icon 24
 - restarting 25
 - starting and stopping the server 23
 - starting multiple instances 24
 - starting and stopping 23
 - stopping 25
- Server Activity Monitor
 - access log 236
 - hints and tips 235
 - network activity statistics 235
 - server activity statistics 235
 - using 234
- server as a caching proxy 11
- server protection
 - access to individual files
 - ACL files, using 215
 - limiting 214
 - using server group files 214
 - activating protection 203
 - address template 205
 - creating protection setups
 - using config and admin forms 207
 - using config file 207
 - using separate setup files 207
 - examples 218
 - identifying setup to requesters 208
 - passing requests 203
 - pointing to a server group file 209
 - pointing to password file 209
 - processing requests 205
 - rules for user names, groups, and addresses 210, 211
 - specifying authentication type 208
 - specifying SSL client authentication parameters 212
 - specifying valid user names, groups, and addresses 210
 - types of 204
 - user name and password 204
- server-side includes
 - format 179
 - preparing to use 178
 - using 178
- ServerID subdirective 87, 208
- ServerInit directive 123
- ServerPriority directive 147
- ServerRoot directive 41
- ServerTerm directive 130
- Service directive 126
- servlet API directives
 - EnableJavaServletSupport 131
 - MaxActiveJavaThreads 132
 - Servlet 133
 - ServletDir 133
 - ServletLog 132
- Servlet directive 133
- ServletDir directive 133
- ServletLog directive 132
- setuperror 115
- shut down server and restart 257
- signatures, digital
 - See digital signature (fingerprint)
- Simple Network Management Protocol 223
 - commands and protocol 224
 - community names 150, 233
 - e-mail address 151, 232
 - enabling SNMP on your system 232
 - object IDs and variable names, secure server MIB 224
 - object IDs for MIB 224
 - overview 223
 - password 150, 233
 - receiving problem reports 151, 232
 - turning support on and off 150, 233, 234
 - turning support on and off, httpd command 234
- SNMP 223
 - commands and protocol 224
 - community names 150, 233
 - e-mail address 151, 232
 - enabling SNMP on your system 232
 - object IDs and variable names, secure server MIB 224
 - object IDs for MIB 224
 - overview 223
 - password 150, 233
 - receiving problem reports 151, 232
 - turning support on and off 150, 233, 234
 - turning support on and off, httpd command 234

- SNMP directive 150
- SNMPCommunityName directive 150
- socksifying a proxy server 13
- SocksServer directive 143
- specifying log paths
 - for access log 155
 - for error log 159
 - for proxy log 161
- specifying valid user names, groups, and addresses 210
- SSL (Secure Sockets Layer)
 - client authentication 256
 - encryption modes 283
 - DES 56 bit 284
 - RC2 284
 - RC4 284
 - Triple DES (EDE) 192 bit 284
 - forms 256
 - overview 244, 255
 - process your request 257
 - security configuration 257
 - setting up 256
 - tunneling 18
 - using 255
 - using with your server 267
- SSL_ClientAuth subdirective 88, 213
- SSLClientAuth directive 90
- SSLMode directive 90
- SSLPort directive 91
- starting a PICS service 194
- starting and stopping the server 23
- stashing passwords 247, 260
- stopping the server 25
- storing files on server 193
- SuffixCaseSense directive 99
- system management directives
 - SNMP 150
 - SNMPCommunityName 150
 - WebMasterEmail 151

T

- Telnet 201
- threads
 - lowering virtual memory use with 144
 - MaxActiveThreads directive 147
 - server use of 144
- time specifiers 36
- timeout directives
 - InputTimeout 116

- timeout directives (*continued*)
 - OutputTimeout 117
 - PersistTimeout 148
 - ScriptTimeout 117
- TRACE method 119
- trusted root keys
 - deleting a key 249
 - designating a key 253
 - managing trusted roots 245
- types of protection
 - address template 205
 - SSL client authentication 212
 - user name and password 204

U

- unknownmethod 113
- updating PICS configuration file 196
- US-Canadian version differences 283
- UseACLs directive 149
- UseMetaFiles directive 149
- user directory directives
 - HomeDir 52
 - UserDir 53
- user name and password protection 204
- UserDir directive 53

V

- valid user names, groups, and addresses,
 - specifying 210
- variable names, SNMP MIB 224
- virtual hosts 19

W

- wais_proxy directive 143
- web site rating criteria 188
- webibm password
 - controlling access
 - password 4, 6
 - webibm password 4, 6
 - webadmin user 6
- WebMasterEmail directive 151
- welcome (home) page
 - creating 9
 - default list 9
 - default, changing 7
- Welcome directive 50

welcome page and directory directives

- AddBlankIcon 41
- AddDirIcon 42
- AddIcon 43
- AddParentIcon 43
- AddUnknownIcon 44
- AlwaysWelcome 45
- DirAccess 45
- DirReadme 46
- DirShowBrackets 46
- DirShowBytes 47
- DirShowCase 47
- DirShowDate 47
- DirShowDescription 48
- DirShowHidden 48
- DirShowIcons 48
- DirShowMaxDescrLength 49
- DirShowMaxLength 49
- DirShowMinLength 49
- DirShowSize 50
- IconPath 50
- UserDir 53
- Welcome 50
- www_browsable files 45

Communicating Your Comments to IBM

IBM Internet Connection Secure Server
Webmaster's Guide
Version 4.2 for OS/2 Warp
Publication No. GC31-8486-00

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

If you are mailing a readers' comment form (RCF) from a country other than the United States, you can give the RCF to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by mail, use the RCF at the back of this book.
- If you prefer to send comments by FAX, use this number:
United States and Canada: **1-800-227-5088**
- If you prefer to send comments electronically, use this network ID:

- IBM Mail Exchange: **USIB2HPD at IBMMAIL**
- IBMLink: **CIBMORCF at RALVM13**
- Internet: **USIB2HPD@VNET.IBM.COM**

Make sure to include the following in your note:

- Title and publication number of this book
- Page number or topic to which your comment applies.

Help us help you!

**IBM Internet Connection Secure Server
Webmaster's Guide
Version 4.2 for OS/2 Warp
Publication No. GC31-8486-00**

We hope you find this publication useful, readable and technically accurate, but only you can tell us! Your comments and suggestions will help us improve our technical publications. Please take a few minutes to let us know what you think by completing this form.

| Overall, how satisfied are you with the information in this book? | Satisfied | Dissatisfied |
|---|--------------------------|--------------------------|
| | <input type="checkbox"/> | <input type="checkbox"/> |

| How satisfied are you that the information in this book is: | Satisfied | Dissatisfied |
|---|--------------------------|--------------------------|
| Accurate | <input type="checkbox"/> | <input type="checkbox"/> |
| Complete | <input type="checkbox"/> | <input type="checkbox"/> |
| Easy to find | <input type="checkbox"/> | <input type="checkbox"/> |
| Easy to understand | <input type="checkbox"/> | <input type="checkbox"/> |
| Well organized | <input type="checkbox"/> | <input type="checkbox"/> |
| Applicable to your task | <input type="checkbox"/> | <input type="checkbox"/> |

Specific Comments or Problems:

Please tell us how we can improve this book:

Thank you for your response. When you send information to IBM, you grant IBM the right to use or distribute the information without incurring any obligation to you. You of course retain the right to use the information in any way you choose.

Name

Address

Company or Organization

Phone No.

Help us help you!
GC31-8486-00



Cut or Fold
Along Line

Fold and Tape

Please do not staple

Fold and Tape



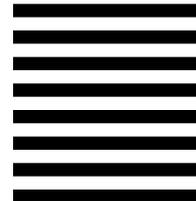
NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

Information Development
Department CGMD
International Business Machines Corporation
PO BOX 12195
RESEARCH TRIANGLE PARK NC 27709-9990



Fold and Tape

Please do not staple

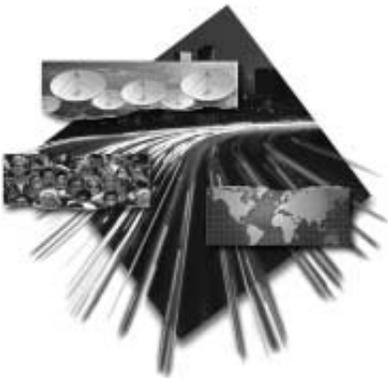
Fold and Tape

GC31-8486-00

Cut or Fold
Along Line



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.



GC31-8486-00

