Blue Coat® Systems
ProxySG® Appliance

Configuration and Management Suite
Volume 2: Proxies and Proxy Services

Version SGOS 5.3.x

Blue Coat®

Contact Information

Blue Coat Systems Inc.
420 North Mary Ave
Sunnyvale, CA 94085-4121

http://www.bluecoat.com/support/contactsupport

http://www.bluecoat.com

For concerns or feedback about the documentation: documentation@bluecoat.com

Document Number: 231-03011
Document Revision: SGOS 5.3.1—08/2008

Contents

Contact Information

Chapter 1: Introduction

About This Book 11
Document Conventions 11
Notes and Warnings 12
About Procedures 12
Illustrations 12
..... 13

Chapter 2: About Management Services

Overview of Management Services 15
Creating a Management Service 16
Managing the HTTP Console 18
Managing the HTTPS Console (Secure Console) 18
Selecting a Keyring 18
Selecting an IP Address 19
Enabling the HTTPS Console Service 19
Managing the SNMP Console 20
Managing the SSH Console 20
Managing the SSH Host 21
Managing SSH Client Keys 22
Managing the Telnet Console 24

Chapter 3: About Proxy Services and Proxies

About Proxy Listeners 28
About Service Attributes and Proxy Services 29
About Multiple Listeners 30
About Proxy Service Groups 31
About Protocol Detection 32

Section A: Managing Proxy Services and Service Groups

Viewing Proxy Services 34
Changing Listener Actions for a Service Group 34
Moving a Service Among Groups 35
Deleting a Service or Service Group 35

Section B: Creating or Editing a Proxy Service

Creating a New Proxy Service 37
Editing an Existing Proxy Service 39
Importing a Service from the Service Library 40

Section C: About Global Options for Proxy Services

Reflecting the Client Source IP when Connecting to Servers 43
Trusting the Destination IP Address Provided by the Client 43
Enabling the ProxySG to Trust the Client-Provided Destination IP Address 44
Tip 44
Managing User Limits 44
Determining Behavior if User Limits are Exceeded 45
Setting User Limits Notifications 46
Viewing Concurrent Users 47
Configuring General Options 47

Section D: About the Bypass List

Adding Static Bypass Entries 49
Using Policy to Configure Dynamic Bypass 50
Notes 50
Configuring Dynamic Bypass 51

Section E: Using Restricted Intercept

Section F: Proxy Services and Listeners

Reference: Access Log Fields 57
Reference: VPM Objects 58
Reference: CPL Policy Configuration for Service Group 58

Chapter 4: Accelerating File Sharing

About the CIFS Protocol 59
About the Blue Coat CIFS Proxy Solution 60
Caching Behavior 61
Authentication 61
Policy Support 61
Access Logging 61
WCCP Support 61
Configuring the ProxySG CIFS Proxy 62
About Windows Security Signatures 62
Intercepting CIFS Services 63
Adding and Configuring New CIFS Services 64
Configuring the CIFS Proxy Options 67
Enabling CIFS Access Logging 69
Reviewing CIFS Protocol Statistics 69
Reference: Equivalent CIFS Proxy CLI Commands 72

Reference: Access Log Fields 73
Reference: CPL Triggers, Properties, and Actions 76
Triggers 76
Properties and Actions: 76

Chapter 5: Managing the Domain Name Service (DNS) Proxy

Configuring the DNS Proxy Service Options 79
Changing the Default DNS Proxy Service to Intercept All IP Addresses on Port 53 79
Creating or Editing a DNS Proxy Service 80
Creating a Resolving Name List 82

Chapter 6: Accelerating the Microsoft Outlook Application (Endpoint Mapper and MAPI Proxies)

Section A: The Endpoint Mapper Proxy Service

About RPC 84
About the Blue Coat Endpoint Mapper Proxy Solution 84
Policy Support 85
Access Logging 85
Configuring Endpoint Mapper Service Options 85
Configuring the Endpoint Mapper Service to Intercept All IP Addresses on Port 135 85
Adding a New Endpoint Mapper Service 86
Reviewing Endpoint Mapper Proxy Statistics 89
Reference: Equivalent Endpoint Mapper Proxy CLI Commands 90
Reference: Access Log Fields 90
Reference: CPL Triggers, Properties, and Actions 91
TCP Tunneling Triggers 91
Properties and Actions 92

Section B: The MAPI Proxy

About MAPI 93
About the Blue Coat MAPI Solution 93
Reducing RPC Messages Across the WAN 94
Maintaining Exchange Connections 95
Supported Servers 95
Access Logging 95
More Conceptual Reference 95
Configuring the ProxySG MAPI Proxy 95
About the MAPI Service 96
Configuring the MAPI Proxy 96
Reviewing MAPI Statistics 97
Reference: Equivalent MAPI Proxy CLI Commands 97
Reference: Access Log Fields 98

Chapter 7: Managing the File Transport Protocol (FTP) Proxy
How Do I...? 101
About FTP 102
Terminology 102
Configuring IP Addresses for FTP Control and Data Connections 102
Configuring the ProxySG for Native FTP Proxy 104
Changing the Default FTP Proxy Service to Intercept 105
Creating or Editing the FTP Service 105
Configuring the FTP Proxy 107
Configuring FTP Clients 108
Configuring FTP Connection Welcome Banners 109
Viewing FTP Statistics 110

Chapter 8: Intercepting and Optimizing HTTP Traffic
How Do I...? 111

Section A: About the HTTP Proxy Service
Configuring the HTTP Proxy Service Options 113
Changing the HTTP Proxy Service to Intercept All IP Addresses on Port 80 113
Creating or Editing an HTTP Proxy Service 114
Configuring IE for Web FTP with an Explicit HTTP Proxy 118

Section B: Configuring the HTTP Proxy Performance
Customizing the HTTP Object Caching Policy 119
About Object Pipelining 120
About HTTP Object Types 121
About Meta Tags 122
About Tolerant HTTP Request Parsing 122
Configuring the Global Defaults on the HTTP Object Caching Policy 123
Selecting an HTTP Proxy Acceleration Profile 127
Using the Normal Profile 128
Using the Portal Profile 128
Using the Bandwidth Gain Profile 128
About HTTP Proxy Profile Configuration Components 129
Configuring the HTTP Proxy Profile 133
Fine-Tuning Bandwidth Gain 135
Allocating Bandwidth to Refresh Objects in Cache 135
Using Byte-Range Support 136
Enabling Revalidate Pragma-No-Cache 138
Interpreting Negative Bandwidth Gain Statistics 138

Section C: Caching Authenticated Data (CAD) and Caching Proxy Authenticated Data (CPAD)
About Caching Authenticated Data (CAD) 141

About Caching Proxy Authenticated Data (CPAD) 142

Section D: Viewing HTTP/FTP Statistics

HTTP / FTP History Statistics 145

Section E: Supporting IWA Authentication in an Explicit HTTP Proxy

Disabling the Proxy-Support Header 149

Chapter 9: Configuing and Managing an HTTPS Reverse Proxy Service

Section A: Configuring the HTTPS Reverse Proxy

Changing the HTTPS Proxy Service to Intercept All IP Addresses on Port 443 152

Creating an HTTPS Reverse Proxy Service 152

Section B: Configuring HTTP or HTTPS Origination to the Origin Content Server

Creating Policy for HTTP and HTTPS Origination 158

Chapter 10: Managing Shell Proxies

About Shell Proxies 159

Customizing Policy Settings for Shell Proxies 160

Conditions 160

Properties 160

Actions 161

Boundary Conditions for Shell Proxies 161

About Telnet Shell Proxies 161

Configuring the Telnet Shell Proxy Service Options 162

Changing the Telnet Shell Proxy Service to Intercept All IP Addresses on Port 23 162

Creating or a New Telnet Shell Proxy Service 163

Viewing Shell History Statistics 166

Chapter 11: Managing a SOCKS Proxy

Configuring the SOCKS Proxy Service Options 168

Changing the SOCKS Proxy Service to Intercept All IP Addresses on Port 1080 168

Creating or Editing a SOCKS Proxy Service 168

Configuring the SOCKS Proxy 170

Using Policy to Control the SOCKS Proxy 171

Using the Permeo PA SOCKS Client with the Blue Coat SOCKS Server 172

Viewing SOCKS History Statistics 174

Viewing SOCKS Clients 174

Viewing SOCKS Connections 175

Viewing SOCKS Client and Server Compression Gain Statistics 175

Chapter 12: Managing the SSL Proxy

Understanding the SSL Proxy 177

Validating the Server Certificate 178

Determining What HTTPS Traffic to Intercept 178
Managing Decrypted Traffic 179
Using the SSL Proxy with ADN Optimization 179

Section A: Intercepting HTTPS Traffic

Setting Up the SSL Proxy in Transparent Proxy Mode 182
Setting Up the SSL Proxy in Explicit Proxy Mode 184
Specifying an Issuer Keyring and CCL Lists for SSL Interception 185
Using Client Consent Certificates 186
Downloading an Issuer Certificate 186

Section B: Configuring SSL Rules through Policy

Using the SSL Intercept Layer 190
Using the SSL Access Layer 192
CPL in the SSL Intercept Layer 194
CPL in the SSL Layer 195
Notes 196

Section C: Viewing SSL Statistics

SSL History Statistics 197
Unintercepted SSL Data 197
Unintercepted SSL Clients 198
Unintercepted SSL Bytes 198

Section D: Advanced Topics

Creating an Intermediate CA using OpenSSL 200
Installing OpenSSL 200
Creating a Root Certificate 200
Modifying the OpenSSL.cnf File 201
Signing the ProxySG CSR 202
Importing the Certificate into the ProxySG 202
Creating an Intermediate CA using Microsoft Server 2003 (Active Directory) 203

Chapter 13: Managing the TCP Tunneling Proxy

TCP-Tunnel Proxy Services Supported 207
Configuring the TCP-Tunnel Proxy Service Options 208
Changing the TCP-Tunnel Proxy Service to Intercept All IP Addresses on All Unattended Ports 208
Creating or Editing a TCP-Tunnel Proxy Service 208

Appendix A: Explicit and Transparent Proxy

About the Explicit Proxy 213
About the Transparent Proxy 213
Creating an Explicit Proxy Server 214

Using the ProxySG as an Explicit Proxy 214
Configuring Adapter Proxy Settings 215
Transparent Proxies 215
Configuring Transparent Proxy Hardware 216
Configuring a Layer-4 Switch 217
Configuring a WCCP-Capable Router 217
Configuring IP Forwarding 217

Glossary

Index

Chapter 1: Introduction

This volume provides information about proxy filters. A *proxy* filters traffic, monitors Internet and intranet resource usage, blocks specific Internet and intranet resources for individuals or groups, and enhances the quality of Internet or intranet user experiences.

A proxy serves as an intermediary between a Web client and a Web server and can require authentication to allow identity-based policy and logging for the client, as discussed in *Volume 4: Securing the Blue Coat ProxySG Appliance™*.

About This Book

This book deals with the following topics:

- Chapter 2: "About Management Services" on page 15
- Chapter 3: "About Proxy Services and Proxies" on page 27
- Chapter 4: "Accelerating File Sharing" on page 59
- Chapter 5: "Managing the Domain Name Service (DNS) Proxy" on page 79
- Chapter 6: "Accelerating the Microsoft Outlook Application (Endpoint Mapper and MAPI Proxies)" on page 83
- Chapter 7: "Managing the File Transport Protocol (FTP) Proxy" on page 101
- Chapter 8: "Intercepting and Optimizing HTTP Traffic" on page 111
- Chapter 9: "Configuing and Managing an HTTPS Reverse Proxy Service" on page 151
- Chapter 10: "Managing Shell Proxies" on page 159
- Chapter 11: "Managing a SOCKS Proxy" on page 167
- Chapter 12: "Managing the SSL Proxy" on page 177
- Chapter 13: "Managing the TCP Tunneling Proxy" on page 207
- Appendix A: "Explicit and Transparent Proxy" on page 213

Document Conventions

The following table lists the typographical and Command Line Interface (CLI) syntax conventions used in this manual.

Table 1–1 Document Conventions

Conventions	Definition
Italics	The first use of a new or Blue Coat-proprietary term.

Table 1–1 Document Conventions (Continued)

`Courier font`	Screen output. For example, command line text, file names, and Blue Coat Content Policy Language (CPL).
`Courier Italics`	A command line variable that is to be substituted with a literal name or value pertaining to the appropriate facet of your network system.
`Courier Boldface`	A Blue Coat literal to be entered as shown.
Arial Boldface	Screen elements in the Management Console.
`{ }`	One of the parameters enclosed within the braces must be supplied
`[ ]`	An optional parameter or parameters.
`|`	Either the parameter before or after the pipe character can or must be selected, but not both.

Notes and Warnings

The following is provided for your information and to caution you against actions that can result in data loss or personal injury:

Note: Information to which you should pay attention.

Important: Critical information that is not related to equipment damage or personal injury (for example, data loss).

WARNING! Used *only* to inform you of danger of personal injury or physical damage to equipment. An example is a warning against electrostatic discharge (ESD) when installing equipment.

About Procedures

Many of the procedures in this volume begin:

- **Select Configuration > *TabName*,** if you are working in the Management Console, or
- **From the (config) prompt,** if you are working in the command line interface (CLI).

Blue Coat assumes that you are logged into the first page of the Management Console or entered into configuration mode in the CLI.

Illustrations

To save space, screen shots illustrating a procedure often have the bottom portion removed, along with the blank space.

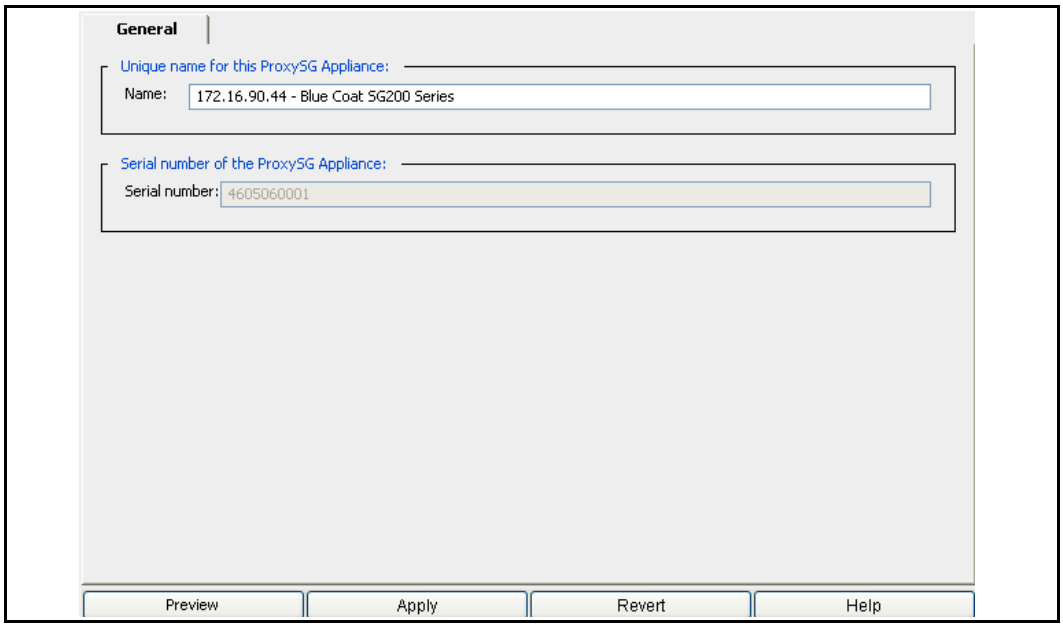


Figure 1–1 **Configuration > General** Tab with Bottom Buttons

- **Preview**: Click this button to view the configuration changes before applying the configuration to the ProxySG. To modify your changes, click **Close** and return to the the tab whose settings you want to modify.
- **Apply**: Click this button to apply unsaved configuration changes to the ProxySG.
- **Revert**: Click this button to revert any unapplied changes to the ProxySG configuration. Changes that previously have been applied to the ProxySG are not affected.
- **Help**: Click this button to view conceptual and procedural documentation about the tab's topic.

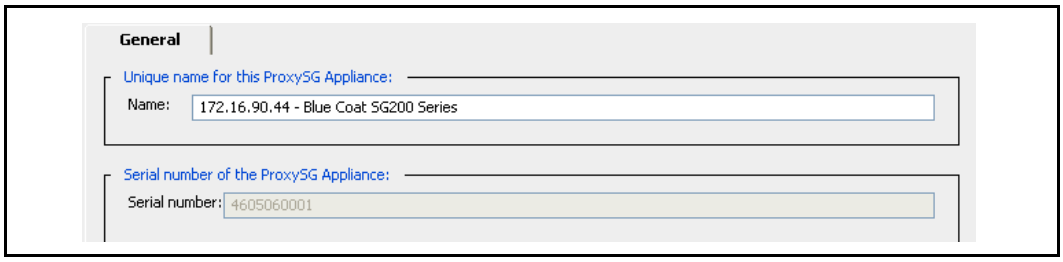


Figure 1–2 **Configuration > General** Tab with Bottom Buttons Removed

Chapter 2: About Management Services

This chapter discusses how to configure and enable managment service listeners.

Topics in this Chapter

This chapter includes information about the following topics:

- "Overview of Management Services" on page 15
- "Creating a Management Service" on page 16
- "Managing the HTTP Console" on page 18
- "Managing the HTTPS Console (Secure Console)" on page 18
- "Managing the SNMP Console" on page 20
- "Managing the SSH Console" on page 20
- "Managing the Telnet Console" on page 24

Overview of Management Services

The ProxySG ships with management services (consoles) that are designed to manage communication with the system:

- HTTP and HTTPS Consoles: These consoles are designed to allow you access to the Management Console. The HTTPS Console is created and enabled by default; the HTTP Console is created by default but not enabled because it is less secure than HTTPS.
- SSH Console: This console is created and enabled by default, allowing you access to the CLI using an SSH client.
- SNMP Console: This console is created by default, but disabled. SNMP listeners set up the UDP and TCP ports the ProxySG uses to listen for SNMP commands.
- Telnet Console: This console not created because the passwords are sent unencrypted from the client to the ProxySG. You must create and enable the console before you can access the appliance through a Telnet client (not recommended).

Table 2–1 Management Services

Management Service	Default Port	Status	Configuration Discussed
HTTPS-Console	8082	Enabled	"Managing the HTTPS Console (Secure Console)" on page 18.
SSH-Console	22	Enabled	"Managing the SSH Console" on page 20

Table 2–1 Management Services (Continued)

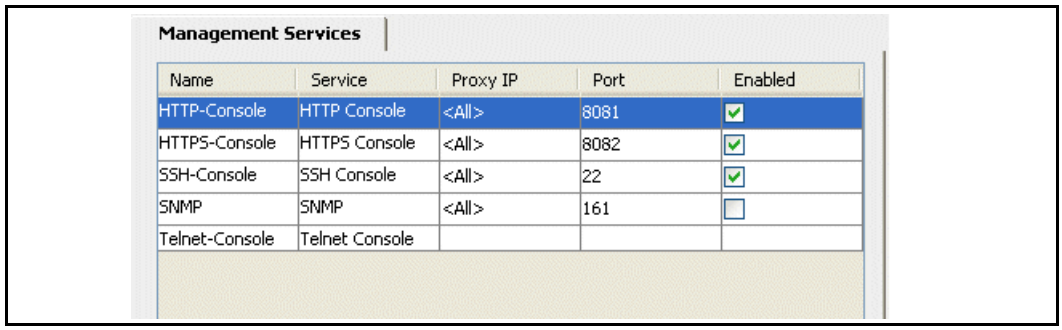
Management Service	Default Port	Status	Configuration Discussed
HTTP-Console	8081	Disabled	"Managing the HTTP Console" on page 18
SNMP	161	Disabled	"Managing the SNMP Console" on page 20
Telnet-Console	—	Not Created	"Managing the Telnet Console" on page 24

Creating a Management Service

Management services are used to manage the ProxySG. As such, bypass entries are ignored for connections to console services.

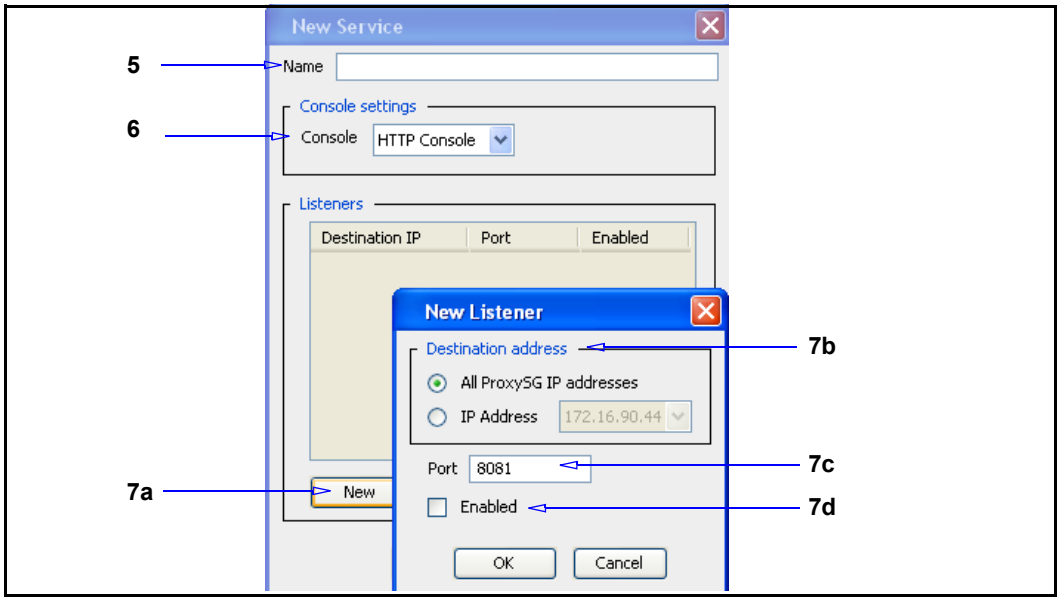
To edit or create a management service:

1. Select **Configuration > Services > Management Services**.



2. To enable or disable a service, select or de-select the **Enable** option.
3. To change other settings on a specific console, highlight the service and click **Edit**.
4. To create a new console service, click **New**.

Note: The HTTP Console is used in this example.



5. Enter a meaningful name in th **Name** field.
6. From the **Console** drop-down list, select the console that is used for this service.
7. Configure the new listener options:
   a. Click **New** to view the **New Listener** dialog. A listener defines the fields where the console service will listen for traffic.
   b. Select a destination option:
      - **All ProxySG IP addresses**—indicates that service listens on all addresses.
      - **IP Address**—indicates that only destination addresses match the IP address.
   c. **Port**–Identifies the port you want this service to listen on. Port 8081 is the default port.
   d. **Enabled**—Select this option to enable the listener.
   e. Click **OK** to close the New Listener dialog.
8. Click **OK** to close the New Service dialog.
9. Click **Apply.**

Related CLI Syntax to Create/Edit a Management Service:

- To enter configuration mode for the service:

```
SGOS(config) management-services
SGOS(config management-services) create {http-console | http-console | snmp | ssh-console | telnet-console} service_name
SGOS(config management-services) edit service_name
```

- The following subcommands are available:

```
SGOS (config service_name) add {all | proxy-ip_address} port_number
{enable | disable}
SGOS (config service_name) disable {all | proxy-ip_address}
port_number
SGOS (config service_name) enable {all | proxy-ip_address} port_number
SGOS (config service_name) exit
SGOS (config service_name) remove {all | proxy-ip_address} port_number
SGOS (config service_name) view
```

Managing the HTTP Console

The default HTTP Console is already configured; you only need to enable it.

You can create and use more than one HTTP Console as long as the IP address and the port do not match the existing HTTP Console settings.

To create a new HTTP Console service or edit an existing one, see "Creating a Management Service" on page 16.

Managing the HTTPS Console (Secure Console)

The HTTPS Console provides secure access to the Management Console through the HTTPS protocol.

You can create multiple management HTTPS consoles, allowing you to simultaneously access the Management Console using any IP address belonging to the ProxySG as well as any of the appliance's virtual IP (VIP) addresses. The default is HTTPS over port 8082.

Creating a new HTTPS Console port requires three steps, discussed in the following sections:

- Selecting a keyring (a key pair and a certificate that are stored together)
- Selecting an IP address and port on the system that the service will use, including virtual IP addresses
- Enabling the HTTPS Console Service

Selecting a Keyring

The ProxySG ships with a default keyring that can be reused with each console that you create. You can also create your own keyrings.

To use the default keyring, accept the default keyring through the Management Console. If using the CLI, the default keyring is automatically used for each new HTTPS Console that is created.To use a different keyring you must edit the console service and select a new keyring using the attribute keyring command.

Note: When using certificates for the HTTPS Console or for HTTPS termination services that are issued by Certificate Signing Authorities that are not well-known, see Chapter 9: "Configuing and Managing an HTTPS Reverse Proxy Service" on page 151.

If you get "host mismatch" errors or if the security certificate is called out as invalid, create a different certificate and use it for the HTTPS Console.

For information on creating a key pair and a certificate to make a keyring, see Chapter 9: "Configuing and Managing an HTTPS Reverse Proxy Service" on page 151.

Selecting an IP Address

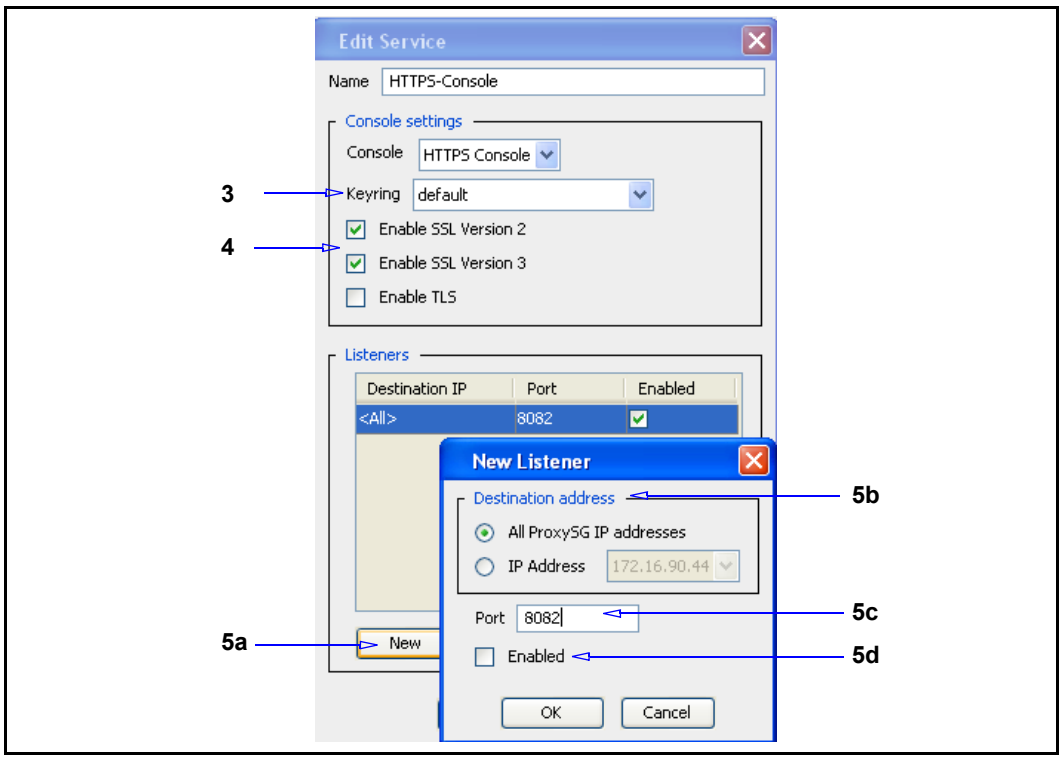
You can use any IP address on the ProxySG for the HTTPS Console service, including virtual IP addresses. To create a virtual IP address, refer to *Volume 5: Advanced Networking*.

Enabling the HTTPS Console Service

The final step in editing or creating an HTTPS Console service is to select a port and enable the service.

To create or edit an HTTPS Console port service:

1. Select **Configuration > Services > Management Services**.
2. Do one of the following:
   - To create a new HTTPS Console service, see "Creating a Management Service" on page 16.
   - To edit the configuration of an existing HTTPS Console service, highlight the HTTPS Console and click **Edit**.



3. From the **Keyring** drop-down list, which displays a list of already-created keyrings on the system, select a keyring. The system ships with a default keyring that is reusable for each HTTPS service.

Note: Two keyrings: configuration-passwords-key keyring an application-key keyring cannot be used for console services.

4. (Optional) Select the appropriate checkboxes to determine the SSL version used for this console.
5. Configure the new listener options:
   a. Click **New** to view the **New Listener** dialog. A listener defines the fields where the console service will listen for traffic.
   b. Select a destination option:
      - **All ProxySG IP addresses**—Indicates that service listens on all addresses.
      - **IP Address**—Indicates that only destination addresses match the IP address.
   c. **Port**—Identifies the port you want this service to listen on. Port 8081 is the default port.
   d. **Enabled**—Select this option to enable the listener.
   e. Click **OK** to close the New Listener dialog.
6. Click **OK** to close the Edit Service dialog.
7. Click **Apply.**

Managing the SNMP Console

There is one disabled SNMP listener defined by default on the ProxySG, which you can delete or enable, as needed. You can also add additional SNMP services and listeners. Enabling SNMP listeners sets up the UDP and TCP ports on which the ProxySG listens for SNMP commands.

To create and enable an SNMP service:

1. Select **Configuration > Services > Management Services**. The Management Services tab displays.
2. Click **Add**. The New Service dialog displays.
3. Follow steps 2–5 in the section titled "Creating a Management Service" on page 16.

Managing the SSH Console

By default, the ProxySG uses Secure Shell (SSH) and password authentication so administrators can access the CLI or Management Console securely. SSH is a protocol for secure remote logon over an insecure network.

When managing the SSH console, you can:

- Enable or disable a version of SSH
- Generate or re-generate SSH host keys

- Create or remove client keys and director keys
- Specify a welcome message for clients accessing the ProxySG using SSHv2.

To create a new SSH Console service or edit an existing one, see "Creating a Management Service" on page 16.

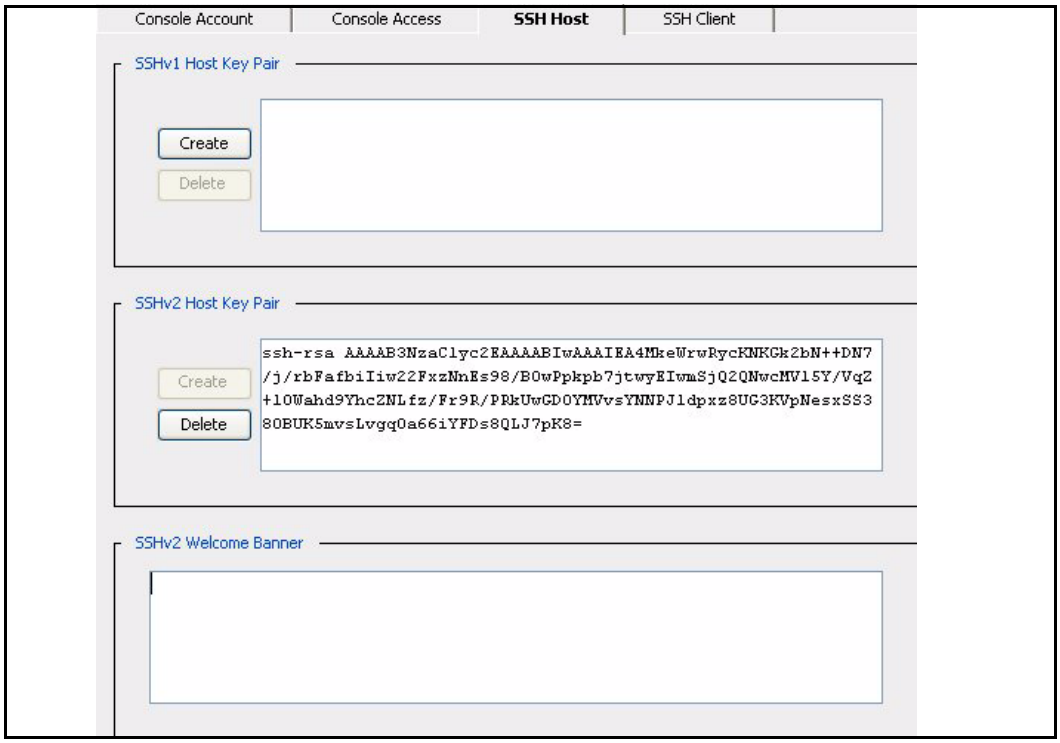
Managing the SSH Host

You can manage the SSH host connection either through the Management Console or the CLI.

To manage the SSH host:

Note: By default, SSHv2 is enabled and assigned to port 22. You do not need to create a new host key unless you want to change the existing configuration. SSHv1 is disabled by default.

1. Select **Configuration > Authentication > Console Access > SSH Host.**



To delete a host key pair:

Click the **Delete** button for the appropriate version of SSH.

The key pair is deleted and that version of SSH is disabled.

Note: If you disable both SSHv1 and SSHv2, you could be locked out of the CLI, requiring you to re-create an SSH key pair using the terminal console. (You can re-create the SSH keys through the Management Console.)

```
SGOS (config ssh-console) create host-keypair {sshv1| sshv2 | <Enter>}
```

To create a host key pair:

Click the **Create** button for the appropriate version of SSH.

The new key pair is created and that version of SSH is enabled. The new key pair is displayed in the appropriate pane.

Note: If you receive an error message when attempting to log in to the system after regenerating the host key pair, locate the ssh **known hosts** file and delete the system's IP address entry.

To create an SSHv2 Welcome Banner:

1. In the **SSHv2 Welcome Banner** field, enter a line of text that will be displayed on the ProxySG when a user attempts to log in to the system. If the message length spans multiple lines, the ProxySG automatically formats the string for multiline capability. The maximum size of the message can be up two thousand characters and can include embedded newlines.

   To delete the welcome banner, remove the text from the **SSHv2 Welcome Banner** field.

2. Click **Apply.**

Managing SSH Client Keys

You can import multiple RSA client keys on the ProxySG to provide public key authentication, an alternative to using password authentication. An RSA client key can only be created by an SSH client and then imported onto the ProxySG. Many SSH clients are commercially available for UNIX and Windows.

Once you create an RSA client key following the instructions of your SSH client, you can import the key onto the ProxySG using either the Management Console or the CLI. (For information on importing an RSA key, see "To import RSA client keys:" on page 23.)

About the OpenSSH.pub Format

Blue Coat supports the `OpenSSH.pub` format. Keys created in other formats will not work.

An `OpenSSH.pub` public key is similar to the following:

```
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAwFI78MKyvL8DrFgcVxpNRHMFKJrBMeBn
2PKcv5oAJ2qz+uZ7hiv7Zn43A6hXwY+DekhtNLOk3HCWmgsrDBE/NOOEnDpLQjBC6t/
T3cSQKZjh3NmBbpE4U49rPduiiufvWkuoEiHUb5ylzRGdXRSNJHxxmg5LiGEiKaoELJfsD
Mc= user@machine
```

The `OpenSSH.pub` format appends a space and a user ID to the end of the client key.

The user ID used for each key must be unique.

Other caveats:

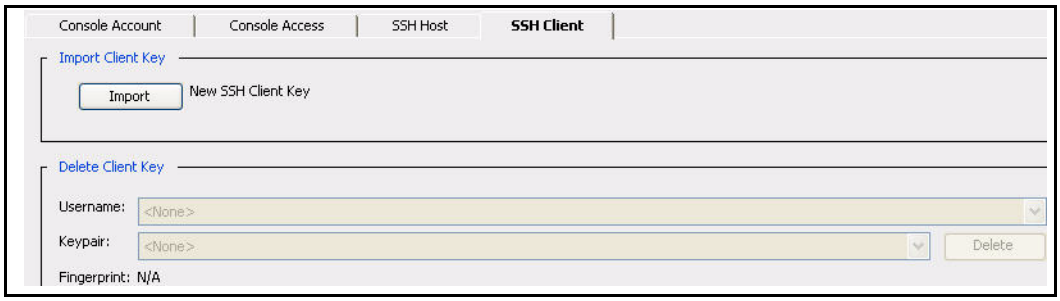
- 1024 bits is the maximum supported key size.
- An `ssh-rsa` prefix must be present.
- Trailing newline characters must be removed from the key before it is imported.

To import RSA client keys:

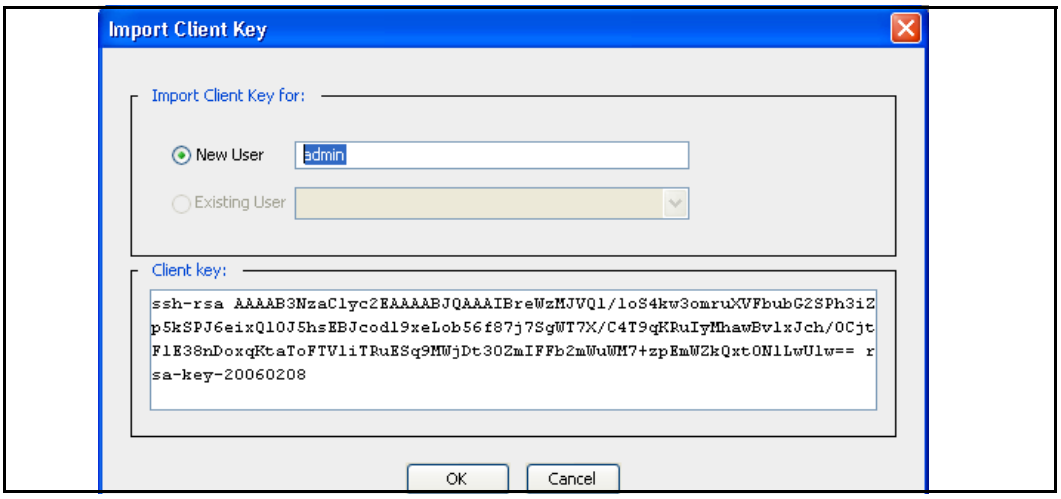
1. From your SSH client, create a client key and copy it to the clipboard.

   Note: The above step must be done with your SSH client. The ProxySG cannot create client keys.

2. Select **Configuration > Authentication > Console Access > SSH Client.**

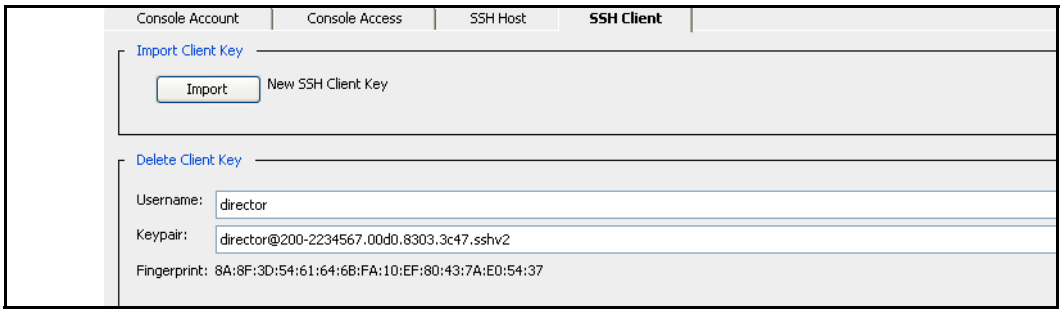


3. Click **Import** to import a new client key.



4. Specify whether the client key is associated with an existing user or a new user, and enter the name.
5. Paste the RSA key that you previously created with an SSH client into the **Client key** field. Ensure that a key ID is included at the end. Otherwise, the import fails.
6. Click **OK**.

The **SSH Client** tab reappears, with the fingerprint (a unique ID) of the imported key displayed.



7. Click **Apply.**

Related CLI Syntax to Manage the SSH Host and Client

```
SGOS (config) ssh-console
```

- The following subcommands are available for managing key pairs and other global options:

```
SGOS (config ssh-console) create host-keypair {sshv1| sshv2 | <Enter>}
SGOS (config ssh-console) delete {client-key username key_id | legacy-client-key key_id | director-client-key key_id | host-keypair {sshv1 | sshv2 | <Enter>}}
SGOS (config ssh-console) inline {client-key <eof> | director-client-key <eof> | sshv2-welcome-banner <eof>}
SGOS (config ssh-console) no sshv2-welcome-banner
SGOS (config ssh-console) view {client-key | director-client-key | host-public-key | sshv2-welcome-banner | user-list | versions-enabled}
```

Managing the Telnet Console

The Telnet console allows you to connect to and manage the ProxySG using the Telnet protocol. Remember that Telnet is an insecure protocol and should be used only in very secure environments. By default, the Telnet Console is not created.

Blue Coat Systems recommends against using Telnet because of the security hole it creates.

Note: If you do enable the Telnet console, be aware that you cannot use Telnet everywhere in the CLI. Some modules, such as SSL, respond with the error message:

```
Telnet sessions are not allowed access to ssl commands.
```

By default a Telnet shell proxy service exists on the default Telnet port (23). Since only one service can use a specific port, you must delete the shell service if you want to create a Telnet console. Be sure to apply any changes before continuing. If you want a Telnet shell proxy service in addition to the Telnet console, you can re-create it later on a different port. For information on the Telnet service, see Chapter 10: "Managing Shell Proxies" on page 159.

To create a new Telnet console service or edit an existing one, see "Creating a Management Service" on page 16.

Note: To use the Telnet shell proxy (to communicate with off-proxy systems) *and* retain the Telnet Console, you must either change the Telnet shell proxy to use a transparent Destination IP address, or change the destination port on either the Telnet Console or Telnet shell proxy. Only one service is permitted on a port. For more information on the Telnet shell proxy, see Chapter 10: "Managing Shell Proxies" on page 159.

Chapter 3: About Proxy Services and Proxies

This chapter discusses proxies, proxy services and service groups. It also describes how to configure a basic proxy service.

Topics in this Chapter

This chapter includes information about the following topics:

- Section A: "Managing Proxy Services and Service Groups" on page 34
- Section B: "Creating or Editing a Proxy Service" on page 37
- Section C: "About Global Options for Proxy Services" on page 43
- Section D: "About the Bypass List" on page 49
- Section E: "Using Restricted Intercept" on page 53
- Section F: "Proxy Services and Listeners" on page 55

For additional information about configuring and managing a specific proxy service see:

- Chapter 4: "Accelerating File Sharing" on page 59
- Chapter 5: "Managing the Domain Name Service (DNS) Proxy" on page 79
- Chapter 6: "Accelerating the Microsoft Outlook Application (Endpoint Mapper and MAPI Proxies)" on page 83
- Chapter 7: "Managing the File Transport Protocol (FTP) Proxy" on page 101
- Chapter 8: "Intercepting and Optimizing HTTP Traffic" on page 111
- Chapter 9: "Configuing and Managing an HTTPS Reverse Proxy Service" on page 151
- Chapter 10: "Managing Shell Proxies" on page 159
- Chapter 11: "Managing a SOCKS Proxy" on page 167
- Chapter 12: "Managing the SSL Proxy" on page 177
- Chapter 13: "Managing the TCP Tunneling Proxy" on page 207

After setting up and enabling the proxy service, the next step is to configure the proxy for your environment. If necessary, you can configure bypass lists for transparent proxy environments. Alternatively, you can specify a list of services that you do want intercepted.

About Proxies and Proxy Services

A *proxy* filters traffic, monitors Internet and intranet resource usage, blocks specific Internet and intranet resources for individuals or groups and enhances the quality of Internet or intranet user experiences.

A proxy serves as an intermediary between a Web client and a Web server and can require authentication to allow identity-based policy and logging for the client, as discussed in *Volume 4: Securing the Blue Coat ProxySG Appliance.*

Proxy services define the ports and addresses where a ProxySG™ listens for incoming requests. A variety of attributes for each service can be defined. Each service can be applied to all IP addresses or limited to a specific set of addresses and port combinations. A number of default services are predefined. Additional services can be defined on other ports.

Blue Coat has two types of services: proxy services, used to communicate with other systems and console services, used to communicate with the ProxySG.

Note: Console services are discussed in Chapter 2: "About Management Services" on page 15.

About Proxy Listeners

A proxy listener is the location where the ProxySG listens for traffic for a specific service. It identifies network traffic based on a destination IP address criterion, a destination port or port range and an action to perform on that traffic. A proxy listener can be identified by any destination IP/subnet and port range, and multiple listeners can be added for each service.

Note: A proxy listener should not be confused with the Default proxy listener, a service that intercepts all traffic not otherwise intercepted by other listeners.

Four settings are available (some settings are not available for some proxy listeners):

- All: This service attribute enables all IP addresses to be intercepted.
- Transparent: This listener type acts on connections without the client or server being aware of it. Only connections to destination addresses that do not belong to the ProxySG are intercepted. This setting requires a bridge, such as that available in the ProxySG; a Layer-4 switch, or a WCCP-compliant router. You can also transparently redirect requests through a ProxySG by setting the workstation's gateway to the appliance IP address.
- Explicit: This listener type is the default and requires software configuration for both browser and service. It sends requests explicitly to a proxy instead of to the origin content servers. Only destinations addresses that match one of the IP addresses on the ProxySG are intercepted.
- Destination IP address or subnet: This listener type ensures that only destination addresses matching the IP address and subnet are intercepted.

  Some software configuration on the ProxySG is also required to allow the appliance to know what traffic to intercept.

Note: For information on understanding explicit and transparent proxies and the configuration requirements, see Appendix A: "Explicit and Transparent Proxy" on page 213.

For a complete list of supported proxy services and listeners, see Section F: "Proxy Services and Listeners" on page 55.

About Service Attributes and Proxy Services

The service attributes define the parameters the ProxySG uses for a particular service.

The following table describes the attributes for a proxy service; however, depending on the protocol, not all attributes are available for each proxy type.

Table 3–1 Service Attributes

Attribute	Description
Authenticate-401	All transparent and explicit requests received on the port always use transparent authentication (cookie or IP, depending on the configuration). This is especially useful to force transparent proxy authentication in some proxy-chaining scenarios.
CA-Cert List	CA Certificate List used for verifying client certificates.
Detect Protocol	Detects the protocol being used. Protocols that can be detected include HTTP, P2P (eDonkey, BitTorrent, FastTrack, Gnutella), SSL, and Endpoint Mapper. For more information on protocol detection, see Chapter 13: "Managing the TCP Tunneling Proxy" on page 207.
Early Intercept	Controls whether the proxy responds to client TCP connection requests before connecting to the upstream server. When early intercept is disabled, the proxy delays responding to the client until after it has attempted to contact the server. If you enable the Detect Protocol attribute, the Early Intercept attribute is selected automatically.
Enable ADN	Controls whether ADN is enabled for a specific service. Enabling ADN does not guarantee the connections are accelerated by ADN. The actual enable decision is determined by ADN routing (for explicit deployment) and network setup (for transparent deployment).
Forward Client Cert	When used with the verify-client attribute, puts the extracted client certificate information into a header that is included in the request when it is forwarded to the OCS. The name of the header is Client-Cert. The header contains the certificate serial number, subject, validity dates and issuer (all as `name=value` pairs). The actual certificate itself is not forwarded.

Table 3–1 Service Attributes (Continued)

Attribute	Description
Optimize Bandwidth	Controls whether to optimize bandwidth usage when connecting upstream using an ADN tunnel.
SSL Versions	Allows you to select which versions of SSL you want to support. The default is to support SSL v2, v3, and TLS. This attribute is available for HTTPS Reverse Proxy.
Verify Client	Requests and validates the SSL client certificate. This attribute is available for HTTPS Reverse Proxy.

About Multiple Listeners

A listener identifies network traffic based on a destination IP address criterion, a destination port or port range and an action to perform on that traffic. Multiple listeners can be defined for a proxy service or console service. Each service has a set of default actions to apply to the traffic identified by the listeners it owns.

The destination IP address of a connection can match multiple proxy service listeners. Multiple matches are resolved using the most-specific match algorithm used by routing devices. A listener is more specific if it has a larger Destination IP subnet prefix. For example, the subnet 10.0.0.0/24 is more specific than 10.0.0.0/16, which is more specific than 10.0.0.0/8.

When a new connection is established, the ProxySG first finds the most specific listener Destination IP. If a match is found, and the destination Port also matches, the connection is then handled by that listener. If the destination Port of the listener with the most specific Destination IP does not match, the next most-specific Destination IP is found; this process continues until either a complete match is found or no more matching addresses are found.

For example, assume the following services were defined:

Table 3–2 Example Configuration for Most Specific Match Algorithm

Proxy Service		Listener	
Service Name	Proxy	Destination IP Address	Port Range
New York Data Center	HTTP	10.167.10.0/24	80
New York CRM	HTTP	10.167.10.2/32	80
HTTP Service	HTTP	<Transparent>	80

An HTTP connection initiated to server 10.167.10.2 could match any of the three listeners in the above table. The most specific match algorithm finds that a listener in the New York CRM service is the most specific and since the destination port of the connection and the listener match, the connection is handled by this service.

The advantage of the most specific match algorithm becomes evident when at some later point another server is added in the New York Data Center subnet. If that server needs to be handled by a different service than the New York Data Center service, a new service with a listener specific to the new server would be added. The administrator does not need to be concerned about rule order in order to intercept traffic to this particular server using the new, most specific service listener.

About Proxy Service Groups

Proxy services are defined on the Proxy Services page (**Configuation > Services > Proxy Services**) and are grouped together into predefined service groups based on the type of traffic they handle. Service groups allow you to:

- Turn on a group of predefined services
- Turn on one listener at a time while maintaining the service grouping
- Intercept traffic on a service group level
- Create custom service groups
- Create and assign a new service to a custom service group when a predefined service group is not sufficient

See Table 3–3, "Service Groups and Services" on page 31 for a complete list of service groups and their associated services.

Note: The HTTPS Reverse Proxy service is also available but not created by default. For information about configuring the HTTPS Reverse Proxy, see Chapter 9: "Configuing and Managing an HTTPS Reverse Proxy Service" on page 151.

Predefined Service Groups and Services

Table 3–3, "Service Groups and Services" lists all service groups and their associated services. All listed service groups are predefined (except Custom Service Groups) and cannot be renamed or deleted. However, you can add a new service to a predefined service group or move a service from one group to another.

Table 3–3 Service Groups and Services

Services Group Name	Services Group Description	Predefined Services
Standard	Services that usually provide content to users	FTP, HTTP, MMS, RTSP

Table 3–3 Service Groups and Services

Services Group Name	Services Group Description	Predefined Services
Intranet	Services that usually constitute intranet traffic	CIFS, Citrix ICA, Endpoint Mapper, LDAP, Lotus Notes, LDP, MS SQL Server, MySQL, NFS, Novell GroupWise, Novell NCP, Oracle, POP3, SMTP, SnapMirror, Sybase SQL
Encrypted	Services that contain encrypted data and therefore should not be ADN-optimized	HTTPS, IMAPS, POP3S
Interactive	Services where the data is interactive	MS Terminal Services, Shell, SSH, Telnet, VNC, X Windows
Reverse-proxy	Placeholder for setting up a reverse proxy deployment	none
Other	Other services the ProxySG can monitor	AOL-IM, MSN IM, Yahoo IM, DNS, SOCKS, default.

About Protocol Detection

Protocol detection identifies HTTP, SSL, Endpoint Mapper and various P2P protocols carried within HTTP CONNECT requests, SOCKS CONNECT requests, and TCP tunnels. On the ProxySG, *protocol detection* can be enabled or disabled for each proxy service manually or it can be implemented using policy. If you set policy for protocol detection, you can enhance granularity by matching on a richer set of conditions than just the specific service; policy always overrides manual settings.

If protocol detection is enabled, the ProxySG inspects the first bytes sent from the client and determines if a corresponding application proxy is available to hand off the connection. For example, an HTTP request identified on a TCP tunnel has full HTTP policy applied to it, rather than just simple TCP tunnel policy. In particular, this means that:

- The request shows up as client protocol **HTTP** rather than **TCP Tunnel**.
- The URL used while evaluating policy is an `http://` URL of the tunneled HTTP request, not a `tcp://` URL of where the tunnel was connecting to.
- Forwarding policy is applied based on the new HTTP request, so the forwarding host selected must support HTTP. A forwarding host of type TCP cannot handle the request and causes the request to be blocked.

Enabling protocol detection helps accelerate the flow of traffic. However, the TCP session must be fully established with the client before either the application proxy or the TCP tunnel proxy contacts the origin server. In some cases, like in the active-mode FTP data connections, enabling protocol detection may cause a delay in setting up the connection.

You can avoid this connection delay either by using a protocol specific proxy such as the FTP proxy or by disabling protocol detection.

If protocol detection is disabled, either in the proxy service check box or through policy, traffic flows over a TCP tunnel without being accelerated by a protocol specific proxy.

Section A: Managing Proxy Services and Service Groups

A service group enables you to turn a set of services on or off collectively. This helps streamline the proxy services configuration process without losing any functionality. This section describes:

"Viewing Proxy Services" on page 34

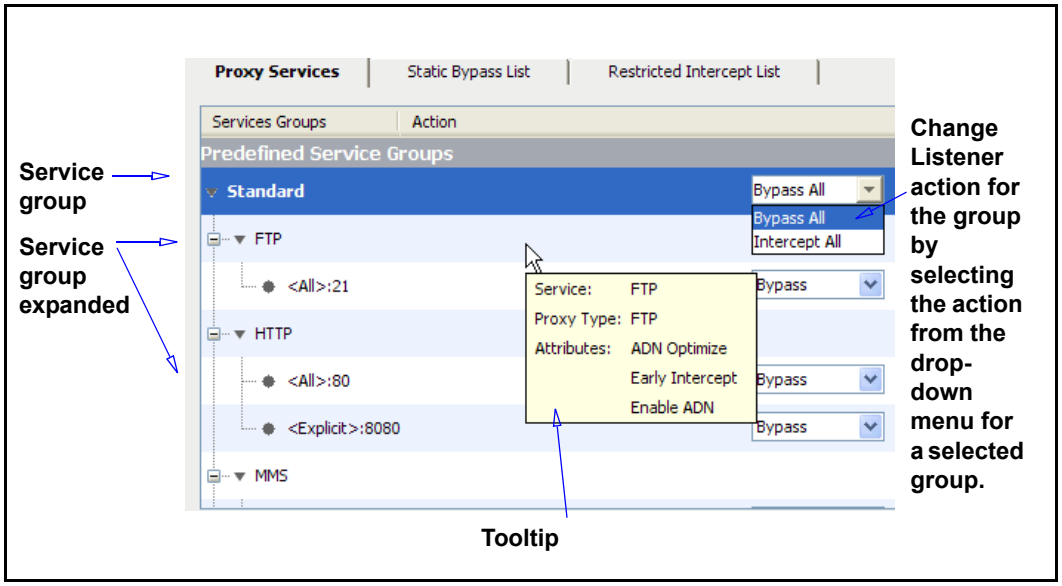
"Changing Listener Actions for a Service Group" on page 34

"Moving a Service Among Groups" on page 35

"Deleting a Service or Service Group" on page 35

Viewing Proxy Services

Click on each service group to show detailed services with their port settings and listener actions. Place the mouse over service to display its tooltip. The tooltip displays service configuration information.



Changing Listener Actions for a Service Group

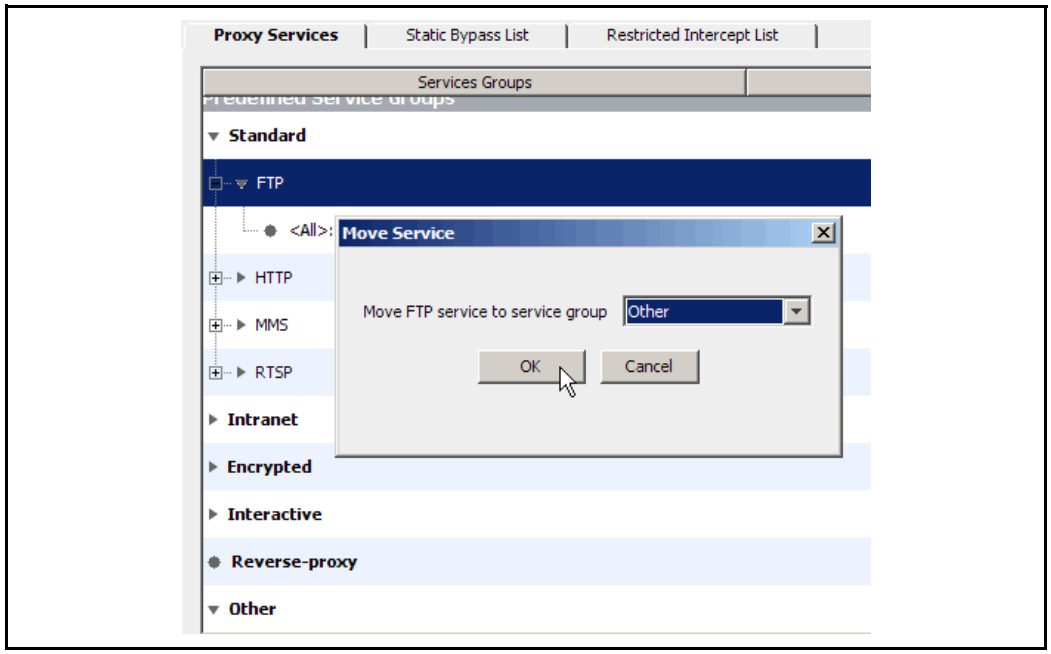
Predefined services are configured, by default, to accept all IP addresses in listener **Bypass** mode. You can change the service listener from **Bypass** to **Intercept** for an entire group by highlighting the service group and selecting **Intercept All** from the **Action** drop-down menu. Or expand a service group and change the service listener for a particular service.

When the listener action is the same for all services in a group the **Action** field for that group displays **Bypass All** or **Intercept All**. If the service listener varies between services within a group the **Action** field for that group displays **Mixed**.

Moving a Service Among Groups

To move a service from one service group to another:

1. From the Management Console, select **Configuration > Services > Proxy Services.**
2. On the **Proxy Services** tab, expand the service group where the service you want to move resides.



3. Select the service you want to move and click **Move Service.** The **Move Service** dialog displays.
4. From the drop-down menu, select the service group into which you want to move the service.
5. Click **OK** to close the dialog; click **Apply.** The service now resides in elected service group.

Deleting a Service or Service Group

To delete a service:

1. On the **Proxy Services** tab, select the service or custom service group you want to delete.
2. Click **Delete**. A prompt appears asking if you want to delete the selected service.
3. Click **Yes**. The selected service or custom service group is deleted.

Note: You can delete a service within a predefined service group but you cannot delete an empty predefined service group itself. However, you can delete the Custom service group if it is empty.

You can add back a default service you deleted from the service library by using the Import Service feature. See "" on page 41.

Relevant CLI Syntax to Create and Configure a Service Group

- To enter configuration mode for the service:

```
SGOS#(config) service-groups
```

- The following subcommands are available:

```
SGOS#(config service-groups){create|bypass-all|delete|intercept-all|
view}
SGOS#(config service-groups) view
```

Section B: Creating or Editing a Proxy Service

This section describes how to create a new or edit an existing proxy service. Only general instructions are given as each specific proxy service has configuration differences.

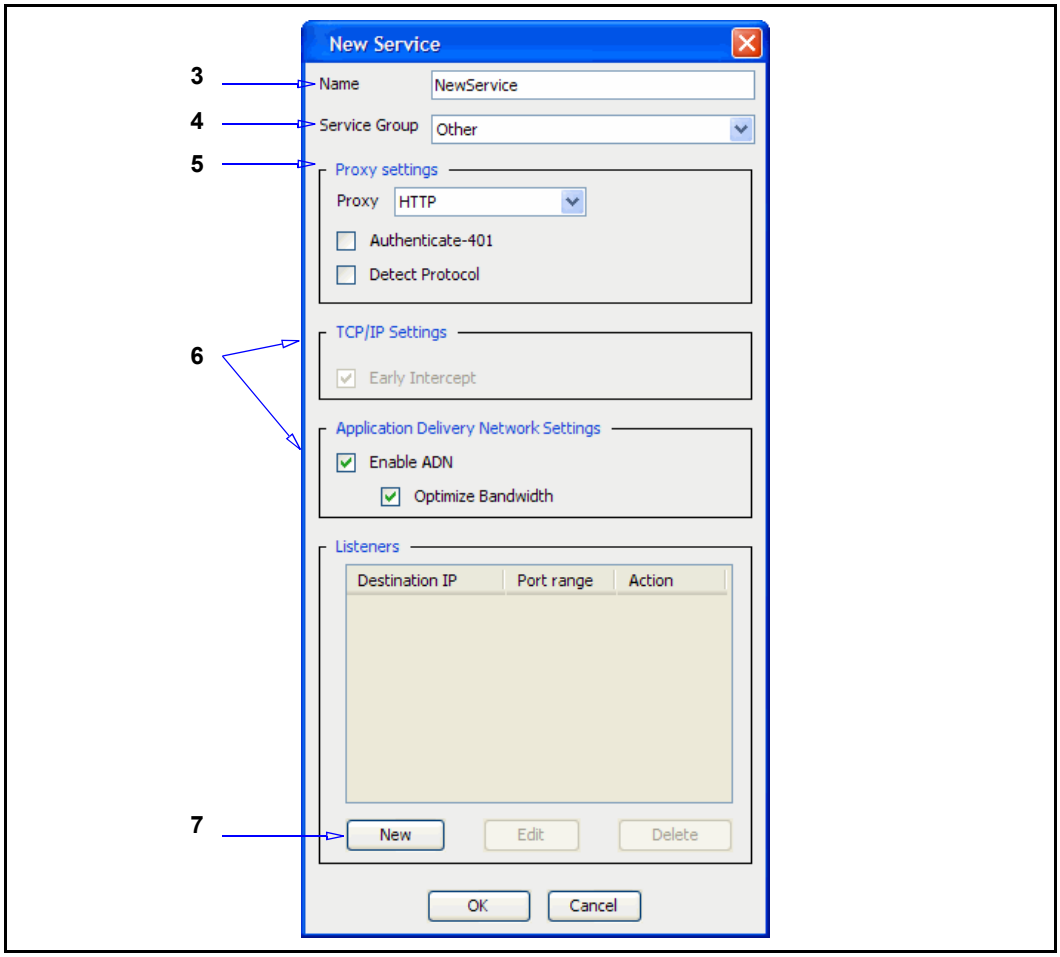
Creating a New Proxy Service

To create a new proxy service:

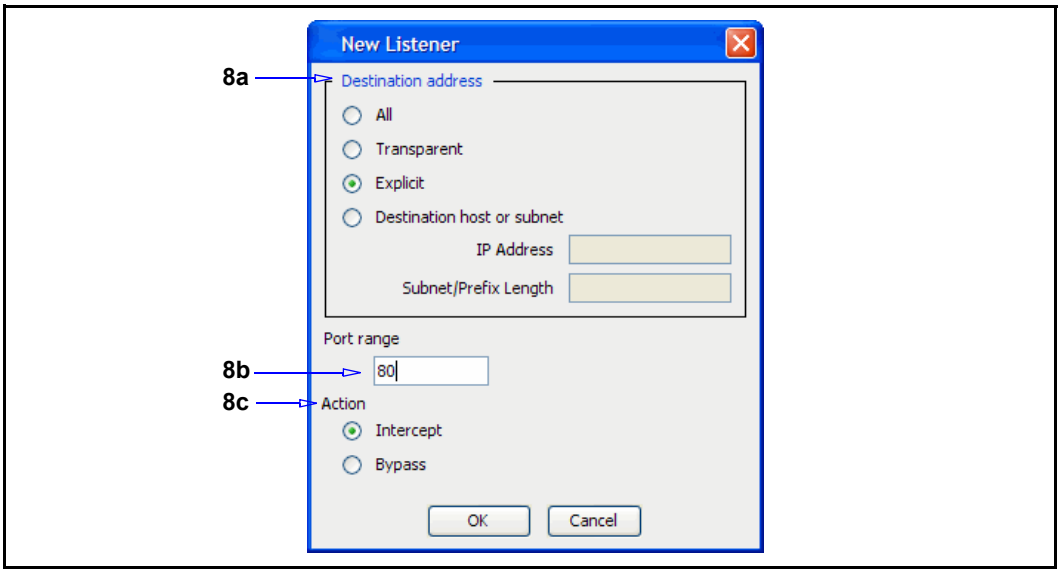
1. From the Management Console, select **Configuration > Services > Proxy Services.** The **Proxy Services** tab displays.
2. At the bottom of the tab, click **New Service**. The **New Service** dialog displays.

Note: If you only want to change the proxy's behavior from bypass (the default) to intercept, go to the **Action** column of the **Proxy Services** pane, select the service whose behavior you want to change, and click **Intercept** from the drop-down list. You do not need to enter New/Edit mode to change this setting.

Section B: Creating or Editing a Proxy Service



3. In the **Name** field, choose a meaningful name for the new proxy service.
4. In the **Service Group** field, select the service group in which you want the service to reside.
5. In the **Proxy Settings** field, select the type of proxy service. The settings below the Proxy field change depending on the kind of proxy you select. (This example is using the TCP-Tunnel proxy.)
6. Enable or clear the options, as appropriate, for the service being set up.
7. To create a new listener, click **New.**



8. Configure the new listener attributes:
   a. Select a **Destination address** from the options.
   b. In the **Port Range** field, enter a single port number or a port range. Port ranges are entered using a - between the start and end ports. For example: `8080-8085`
   c. Select the default action for the service: **Bypass** tells the service to ignore any traffic matching this listener. Intercept configures the service to intercept and proxy the associated traffic.
   d. Click **OK**.
9. Click **Apply**.

See Also

- "Managing Proxy Services and Service Groups"
- "Moving a Service Among Groups"
- "Editing an Existing Proxy Service"
- "Importing a Service from the Service Library"

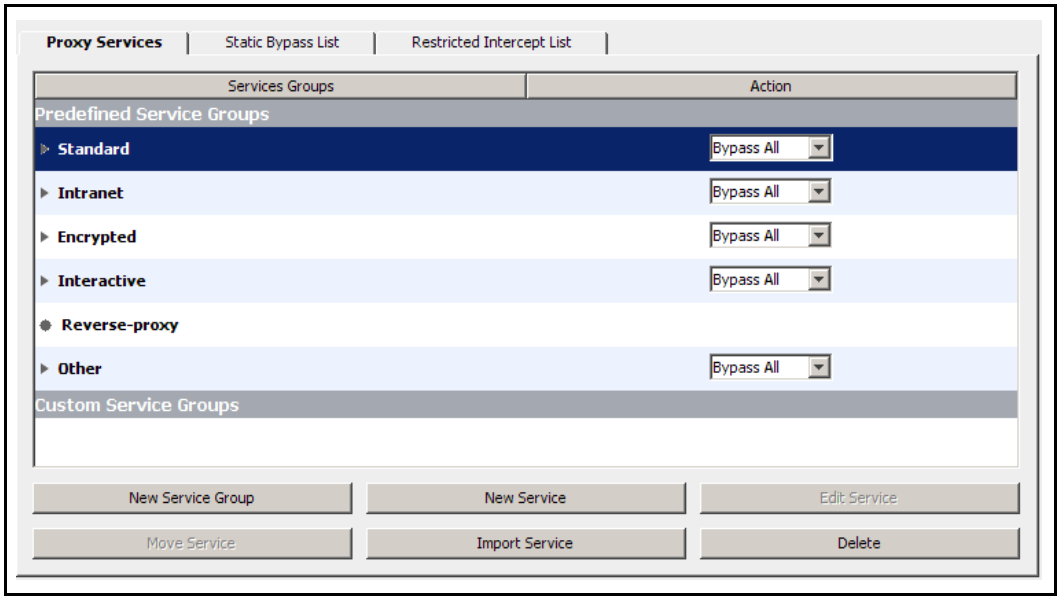
Editing an Existing Proxy Service

To edit an existing proxy service:

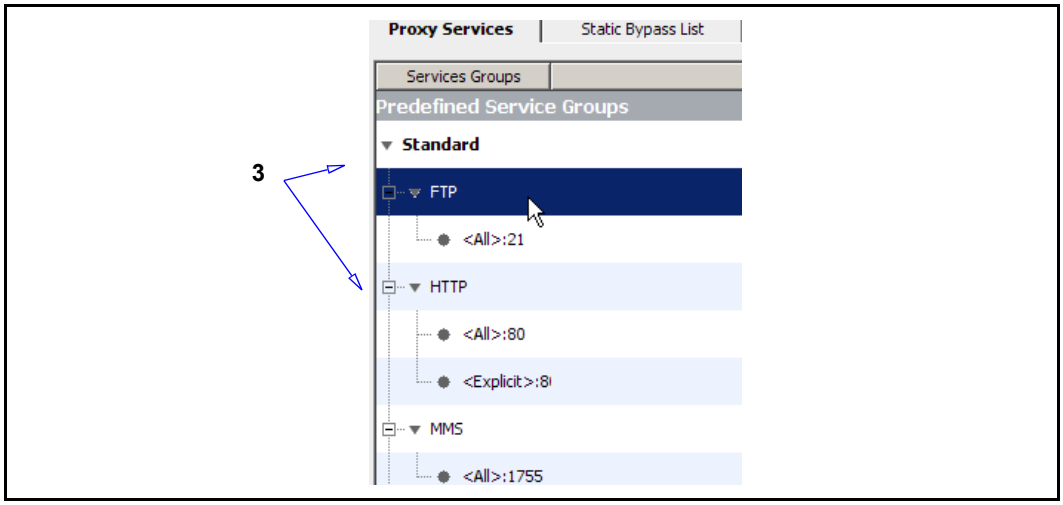
Note: In this example, the **FTP Edit Service** dialog displays. Edit Service dialogs differ depending on which service you select.

---

1. From the Management Console, select **Configuration > Services > Proxy Services.**



2. Scroll the list of service groups and click a service group to expand.



3. Select the service whose configuration you want to edit and then click **Edit Service** at the bottom of the page. The **Edit Service** dialog displays.
4. Edit the fields and click **OK**.

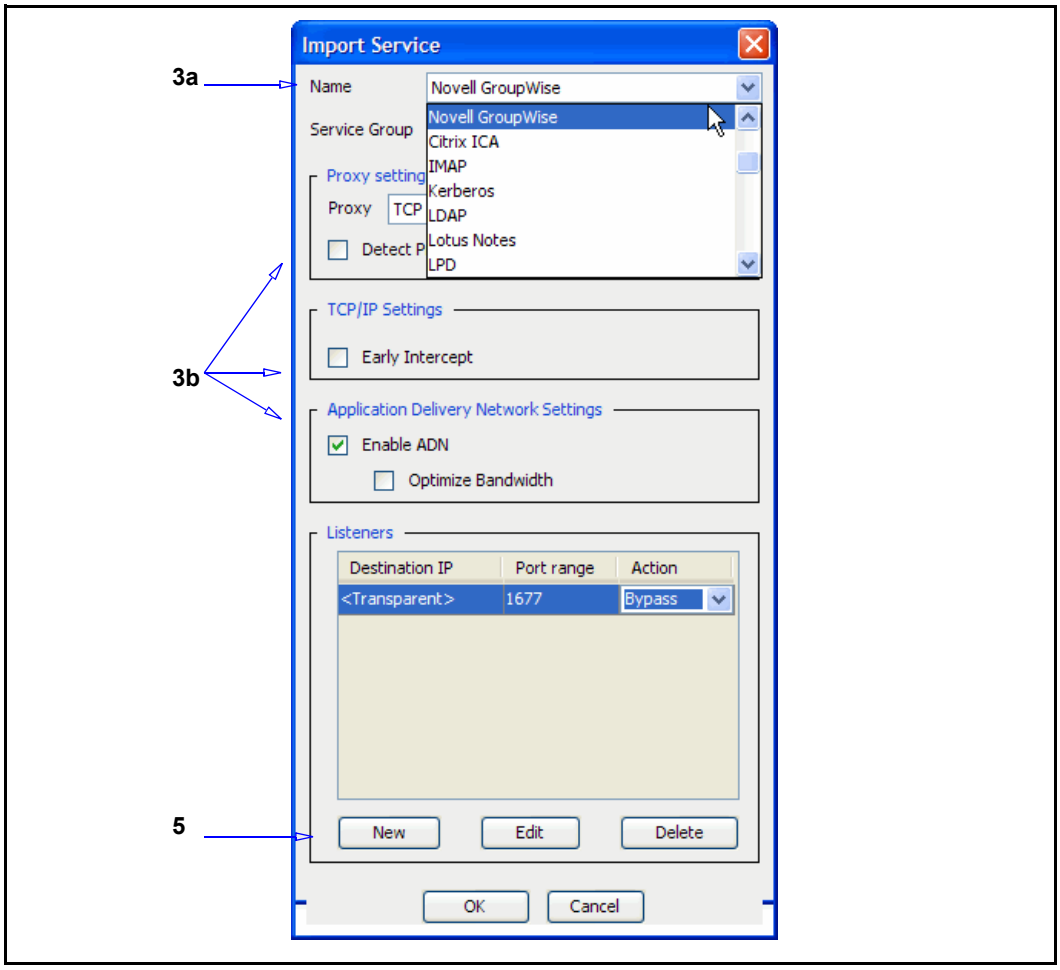
Importing a Service from the Service Library

If needed, you can import a service from the service library. This is useful if you delete a default service and want to add it back. The list of default services on the Proxy Services tab is identical to those in the services library.

To import a service from the service library:

1. From the Management Console, select **Configuration > Services > Proxy Services.**
2. On the **Proxy Services** tab, click **Import Service**. The **Import Service** dialog displays.

Section B: Creating or Editing a Proxy Service



3. Configure the import service options:
   a. In the **Name** field select the service you want to import from the drop-down menu.
   b. All other settings adjust automatically to the service's default values. Perform changes if needed.
   c. Click **New** to configure a new listener or **Edit** to modify existing listener settings.
   d. Click **OK**.
4. Click **Apply**.

Relevant CLI Syntax to Create/Edit a Proxy Service:

- To enter configuration mode for the service:

  ```
  SGOS#(config) proxy-services
  ```

- To create a new service:

Section B: Creating or Editing a Proxy Service

```
SGOS#(config proxy-services) create {aol-im | cifs | dns | endpoint-
mapper | ftp | http | https-reverse-proxy | mms | msn-im | rtsp | socks
| ssl | tcp-tunnel | telnet | yahoo-im} service-name service-group
SGOS#(config proxy-services) edit service-name
```

- The following subcommands are available:

```
SGOS#(config service-name) add {transparent | explicit | all |
ip_address | ip_address/subnet-mask} {port | first_port-last_port}
[intercept | bypass]
SGOS#(config service-name) attribute {authenticate-401 | adn-optimize
| ccl | cipher-suite | detect-protocol | early-intercept | forward-
client-cert | keyring | ssl-versions | use-adn | verify-client}
SGOS#(config service-name) bypass {transparent | explicit | all |
ip_address | ip_address/subnet-mask} {port | first_port-last_port}
SGOS#(config service-name) group service-group
SGOS#(config service-name) intercept {transparent | explicit | all |
ip_address | ip_address/subnet-mask} {port | first_port-last_port}
SGOS#(config service-name) proxy-type proxy-type
SGOS#(config service-name) remove {transparent | explicit | all |
ip_address | ip_address/subnet-mask} {port | first_port-last_port}
SGOS#(config service-name) view
```

- To import a predefined service into a service group from the services library:

```
SGOS#(config proxy-services) import predefined service-name
```

- The following subcommands are available:

```
SGOS#(config proxy-services) import predefined service-name overwrite
```

Section C: About Global Options for Proxy Services

Blue Coat provides three global option settings for proxy services:

- Reflecting the client's source IP when connecting to servers

  The Reflect Client IP option determines how the client IP address is presented to the origin server for all requests. Enable this option when you want the client's IP address sent to the origin server instead of the ProxySG's IP address.

  Note: The Reflect Client IP option is only supported in transparent ProxySG deployments.

- Trusting the Destination IP Address Provided by the Client

  If, in your environment, a client sometimes provides a destination IP address that the ProxySG appliance cannot determine, you can configure the ProxySG to allow that IP address and not do a DNS lookup. This can improve performance (but potentially cause a security issue).

- Managing User Limits

  If you have more users going through the system than is allowed by the model license, you can configure overflow behavior to be queued or to bypass the ProxySG.

Reflecting the Client Source IP when Connecting to Servers

You can globally turn on the Reflect Client IP option for all services that will be intercepted. To turn on the Relfect Client IP option for only a few services, first enable this option globally and then create policy to disable the Reflect Client IP option for the exceptions. Or, disable the option globally and create policy to enable it.

Trusting the Destination IP Address Provided by the Client

You can configure the ProxySG appliance to trust a client-provided destination IP address in transparent proxy deployments where:

- The DNS configuration on the client is correct, but is not correct on the ProxySG.
- The client obtains the destination IP address using WINS or DNS imputing on the ProxySG is not configured correctly. In these cases, the appliance cannot obtain the destination IP address to serve the client request.

You can use the client-provided destination IP address with transparent proxy environments that use HTTP, native FTP, WebFTP, HTTPS, or streaming.

The ProxySG cannot trust the client-provided destination IP address in the following situations:

- The ProxySG receives the client requests in an explicit proxy deployment.
- The ProxySG has a forwarding rule configured for the request.
- The ProxySG has a SOCKS gateway rule configured for the request.
- The ProxySG has ICP enabled for the request.
- The ProxySG has policy that rewrites the server URL

Note: If you are using an Application Delivery Network (ADN), this setting is enforced on the concentrator proxy through the **Configuration > Proxy Settings> General** tab. For more information, refer to *Volume 5: Advanced Networking*.

Enabling the ProxySG to Trust the Client-Provided Destination IP Address

Defaults:

- Proxy Edition: the ProxySG appliance does not trust a client-provided destination IP address.
- MACH5 Edition: The ProxySG appliance trusts client-provided destination IP addresses.

You can change this default through the Management Console (**Configuration > Proxy Settings > General)** the CLI, or through policy. If you use policy, be aware that it overrides any other configuration. For information about using the `trust_destination_ip(yes|no)` property, refer to *Volume 10: Blue Coat SG Appliance Content Policy Language Guide*.

Note: For the MACH5 edition, the ProxySG allows the client-provided destination IP address by default.

For information about enabling the ProxySG to allow the client-provided destination IP address, see "Configuring General Options" on page 47.

Tip

If a client gives the destination address of a blocked site but the hostname of a non-blocked site, the ProxySG connects to the destination address. This might allow clients to bypass the configured appliance security policy.

Managing User Limits

If your ProxySG is in demo or trial mode, an unlimited number of users can have connections processed by the system at the same time.

After a permanent model license has been applied to the system, the license controls the number of active users who can have connections processed by the system at the same time The number of users depends on whether ADN is enabled and on the hardware model.

Only unique IP addresses of connections intercepted by proxy services are counted toward the user limit. The number of permitted users is illustrated in the table below.

Table 3–4 Hardware Models and Licensed Users

Hardware Model	Number of Users (Without ADN Enabled)	Number of Users (With ADN Enabled)
210-5	30	10
210-10	150	50
210-25	Not License Limited	Not License Limited
510-5	200	50
510-10	500	125
510-20	1200	300
510-25	Not License Limited	Not License Limited
810-5	2500	500
810-10	3500	700
810-20	5000	1000
810-25	Not License Limited	Not License Limited
8100-5	Not License Limited	Not License Limited
8100-10	Not License Limited	Not License Limited
8100-20, 8100-20-DC	Not License Limited	Not License Limited

Determining Behavior if User Limits are Exceeded

If you have more user connections going through the system simultaneously than is allowed by the model license, you can configure overflow behavior in the following ways:

- Bypassing the system: All connections exceeding the maximum are passed through the system without processing.
- Queuing connections: All connections exceeding the maximum are queued, waiting for another connection to drop off.
- Not enforcing the licensed-user limit: This is the default value.

Note: SGOS 5.2.1 and later has two options only: **Queue** and **Bypass**. **Queue** is the default.

To set the preferred behavior, see "Configuring General Options" on page 47.

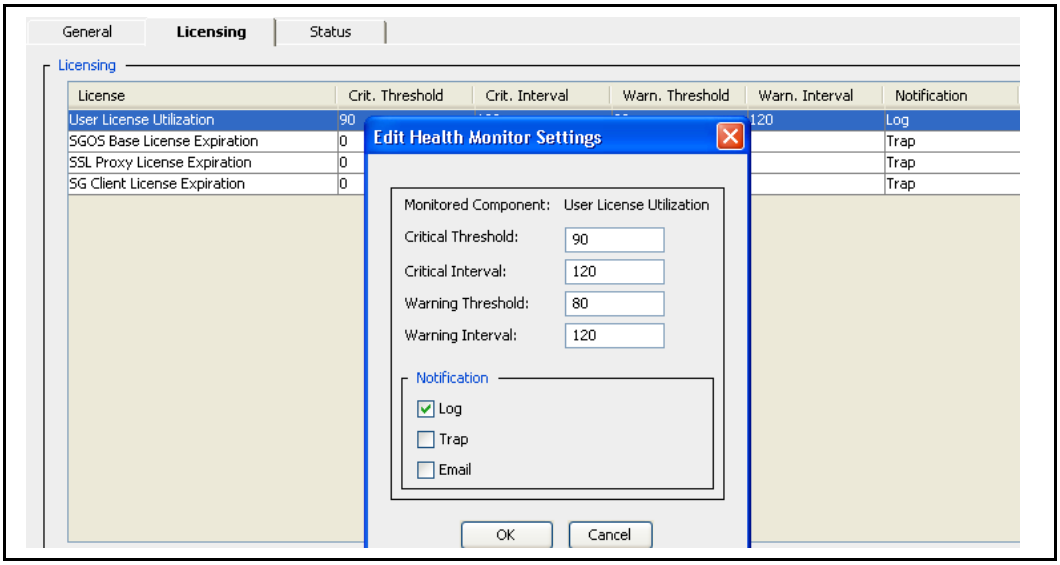
Setting User Limits Notifications

You can set and monitor user limits of the model license through the **Maintenance > Health Monitoring > Licensing** tab, including setting thresholds (in percentages) to be notified if the user limits are nearing the upper user limits.

Note: You can use the **Statistics > Health Monitoring > Licensing** tab to view licensing metrics, but you cannot make changes to the threshold values from that tab. To change the thresholds, use the **Maintenance > Health Monitoring > Licensing** tab.

To view licensing metrics and set user limits notifications:

1. Click **Maintenance > Health Monitoring > Licensing.**



2. Select the license to edit.
3. Click **Edit**.
4. Modify the threshold values. Note that the thresholds represent the percentage of license utilization.
   a. To change the critical threshold, enter a new value in the Critical Threshold field.
   b. To change the critical interval, enter a new value in the Critical Interval field.
   c. To change the warning threshold, enter a new value in the Warning Threshold field.
   d. To change the warning interval, enter a new value in the Warning Interval field.
5. Select the notification settings.
   - **Log** adds an entry to the Event log.

- **Trap** sends an SNMP trap to all configured management stations.
- **Email** sends an e-mail to the addresses listed in the Event log properties.

6. Click **OK**.
7. Click **Apply**.

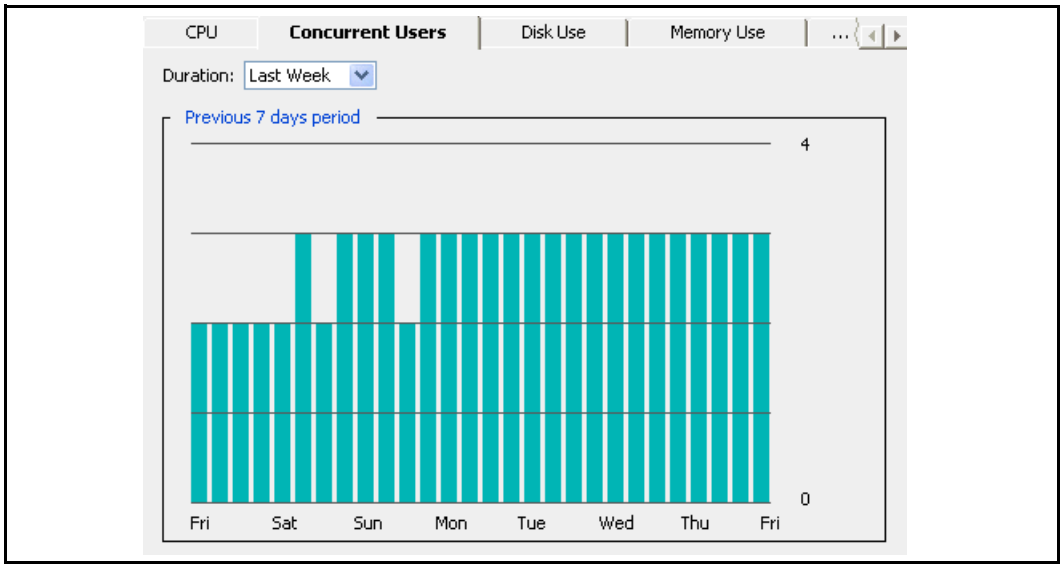
For information about licensing metrics, refer to *Volume 9: Managing the ProxySG*.

Related CLI Syntax to Manage User Limits

```
SGOS#(config) alert notification license-utilization users {email | log | trap | none}
SGOS#(config) alert threshold license-utilization users warn-threshold warn-interval crit-threshold crit-interval
```

Viewing Concurrent Users

View a snapshot of intercepted, concurrent users by selecting the **Statistics > System > Resources > Concurrent Users** tab. The tab shows user connections going through the ProxySG appliance for the last 60 minutes, day, week, month, and year. Only unique IP addresses of connections intercepted by proxy services are counted toward the user limit.

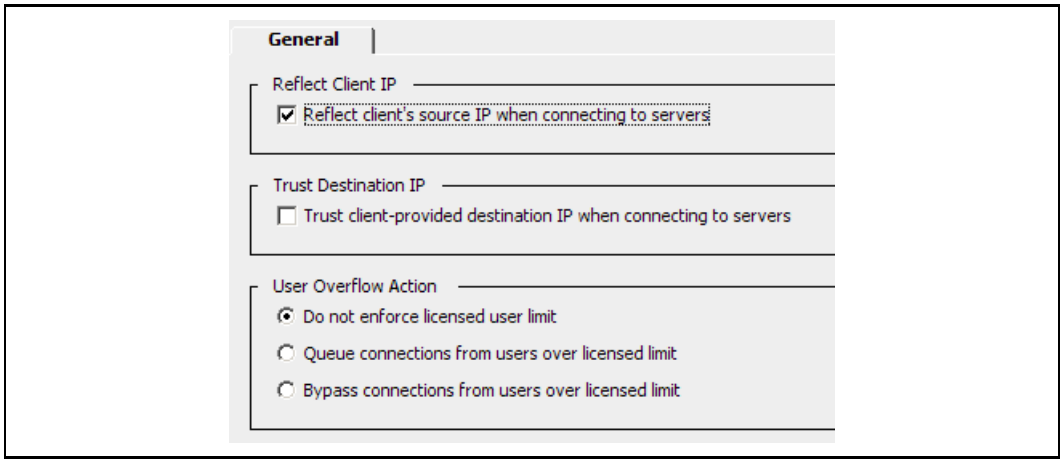


Configuring General Options

You can configure the Reflect Client IP and Trust Destination IP options and the behavior if user limits are exceeded on the **Configuration > Proxy Settings > General** tab. For detailed information about these options, see Section C: "About Global Options for Proxy Services" on page 43.

To configure General options:

1. Click **Configuration > Proxy Settings > General**.



2. Select or clear the **Reflect client's source IP when connecting to servers** option.
3. Select or clear the **Trust client-provided destination IP when connecting to servers** option.
4. Click the radio button for the **User Overflow Action** you prefer when the licensed-user limits are exceeded. By default, for SGOS 5.2.2 and later, **none** is the default.

   Note: If you set the **User Overflow Action** to **none** and exceed the licensed-user limits, the ProxySG health changes to **CRITICAL**.

5. Click **Apply**.

Related CLI Syntax to Manage Reflect Client IP and User Limit Notifications

```
SGOS#(config) general
SGOS#(config general) reflect-client-ip {enable | disable}
SGOS#(config general) trust-destination-ip {enable | disable}
SGOS#(config general) user-overflow-action {bypass | none | queue}
```

See Also

- "About Global Options for Proxy Services"
- "Reflecting the Client Source IP when Connecting to Servers"
- "Trusting the Destination IP Address Provided by the Client"
- "Managing User Limits"

Section D: About the Bypass List

The bypass list contains IP addresses / subnet masks of client and server workstations. Used only in a transparent proxy environment, the bypass list allows the ProxySG to skip processing requests sent from specific clients to specific servers. The list allows traffic between protocol incompliant clients and servers to pass through the ProxySG without a disruption in service.

Note: This prevents the appliance from enforcing any policy on these requests and disables any caching of the corresponding responses. Because bypass entries bypass Blue Coat policy, use bypass sparingly and only for specific situations.

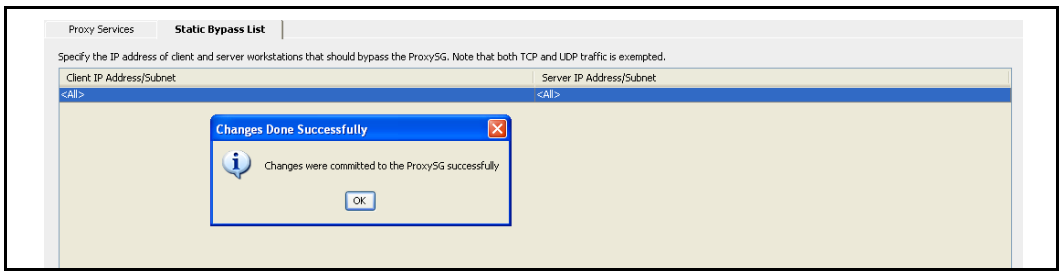
Adding Static Bypass Entries

You can add entries to prevent the requests from specified systems from being intercepted by the ProxySG.

Note: Dynamic bypass cannot be configured through the Management Console. It can only be configured through policy or the CLI. For more information, see " Using Policy to Configure Dynamic Bypass" on page 50.

To add static bypass entries:

1. Click **Configuration > Services > Proxy Services > Static Bypass List.**
2. Click **New** to create a new list entry; click **Edit** to modify a list entry.



3. Fill in the fields:
   a. Select a source IP address from the drop-down list or choose <**All**>. Add the subnet mask.
   b. Select a destination IP address from the drop-down list or choose <**All**>. Add the subnet mask.
4. Click **OK**; click **Apply.**

Relevant CLI Syntax to Manage Static Bypass Entries

- To configure the service:

```
SGOS#(config) proxy-services
SGOS#(config proxy-services) static-bypass
```

- The following subcommands are available:

```
SGOS#(config static-bypass) add {all | client_ip_address |
client_ip_address/subnet-mask} {all | server_ip_address |
server_ip_address/subnet-mask}
SGOS#(config static-bypass) remove {all | client_ip_address |
client_ip_address/subnet-mask} {all | server_ip_address |
server_ip_address/subnet-mask}
SGOS#(config static-bypass) view {filter {* | all | client_ip_address
| client_ip_address/subnet-mask} {* | all | server_ip_address |
server_ip_address/subnet-mask} | <Enter>}
```

Using Policy to Configure Dynamic Bypass

Dynamic bypass, available through policy, can automatically compile a list of response URLs that return various kinds of errors.

Note: Because bypass entries bypass Blue Coat policy, the feature should be used sparingly and only for specific situations.

Dynamic bypass keeps its own (dynamic) list of which connections to bypass, where connections are identified by both source and destination. Dynamic bypass can be based on any combination of policy triggers. In addition, some global settings can be used to selectively enable dynamic bypass based on specific HTTP response codes. After an entry exists in the dynamic bypass table for a specific source / destination IP pair, all connections from that source IP to that destination IP are bypassed in the same way as connections that match against the static bypass list.

For a configured period of time, further requests for the error-causing URLs are sent immediately to the origin content server (OCS), bypassing the ProxySG. The amount of time a dynamic bypass entry stays in the list and the types of errors that cause the ProxySG to add a site to the list, as well as several other settings, are configurable from the CLI.

Once the dynamic bypass timeout for a client and server IP address entry has ended, the ProxySG removes the entry from the bypass list. On the next client request for the client and server IP address, the ProxySG attempts to contact the OCS. If the OCS still returns an error, the entry is once again added to the local bypass list for the configured dynamic bypass timeout. If the entry does not return an error, entries are again added to the dynamic list and not the local list.

Notes

- Dynamic bypass entries are lost when the ProxySG is restarted.
- No policy enforcement occurs on client requests that match entries in the dynamic or static bypass list.
- If a site that requires forwarding policy to reach its destination is entered into the bypass list, the site is inaccessible.

Configuring Dynamic Bypass

Dynamic bypass is disabled by default. Enabling and fine-tuning dynamic bypass is a two-step process:

- Set the desired dynamic bypass timeout and threshold parameters.
- Use policy (recommended) or the CLI to enable dynamic bypass and set the types of errors that cause dynamic bypass to add an entry to the bypass list.

Adding Dynamic Bypass Parameters to the Local Bypass List

The first step in configuring dynamic bypass is to set the `server-threshold, max-entries, or timeout` values.

Note: This step is optional because the ProxySG uses default configurations if you do not specify them. Use the default values unless you have specific reasons for changing them. Contact Blue Coat Technical Support for detailed advice on customizing these settings.

- The `server-threshold` value defines the maximum number of client entries before the ProxySG consolidates client–server pair entries into a single server entry that then applies to all clients connecting to that server. The range is 1 to 256. The default is 16. When a consolidation occurs, the lifetime of the consolidated entry is set to the value of `timeout`.
- The `max-entries` defines the maximum number of total dynamic bypass entries. The range is 100 to 50,000. The default value is 10,000. When the number of entries exceeds the `max-entries` value, the oldest entry is replaced by the newest entry.
- The `timeout` value defines the number of minutes a dynamic bypass entry can remain unreferenced before it is deleted from the bypass list. The range is 1 to 86400. The default value is 60.

Enabling Dynamic Bypass and Specifying Triggers

Enabling dynamic bypass and specifying the types of errors that causes a URL to be added to the local bypass list are done with the CLI. You cannot use the Management Console.

Using policy to enable dynamic bypass and specify trigger events is better than using the CLI, because the CLI has only a limited set of responses. For information about available CLI triggers, refer to the *Volume 11: Command Line Interface Reference*. For information about using policy to configure dynamic bypass, refer to the *Volume 10: Content Policy Language Guide*.

Bypassing Connection and Receiving Errors

In addition to setting HTTP code triggers, you can enable connection and receive errors for dynamic bypass.

If `connect-error` is enabled, any connection failure to the origin content server (OCS), including timeouts, inserts the OCS destination IP address into the dynamic bypass list.

If `receive-error` is enabled, when the cache does not receive an HTTP response on a successful TCP connection to the OCS, the OCS destination IP address is inserted into the dynamic bypass list. Server timeouts can also trigger `receive-error`. The default timeout value is 180 seconds, which can be changed (refer to *Volume 1: Getting Started*).

Related CLI Syntax to Enable Dynamic Bypass and Trigger Events

- To enter configuration mode for the service:

```
SGOS#(config) proxy-services
SGOS#(config proxy-services) dynamic-bypass
```

- The following subcommands are available:

```
SGOS#(config dynamic-bypass) {enable | disable}
SGOS#(config dynamic-bypass) max-entries number
SGOS#(config dynamic-bypass) server-threshold number
SGOS#(config dynamic-bypass) trigger {all | connect-error | non-http |
receive-error | 400 | 403 | 405 | 406 | 500 | 502 | 503 | 504}
SGOS#(config dynamic-bypass) timeout minutes
#(config dynamic-bypass) no trigger {all | connect-error | non-http |
receive-error | 400 | 403 | 405 | 406 | 500 | 502 | 503 | 504}
SGOS#(config dynamic-bypass) clear
SGOS#(config dynamic-bypass) view
```

Section E: Using Restricted Intercept

By default, all clients and servers evaluate the entries in Proxy Services (**Configuration > Services > Proxy Services**) where the decision is made to intercept or bypass a connection. To restrict or reduce the clients and servers that can be intercepted by proxy services, use the Restricted Intercept List. The Restricted Intercept List is useful in a rollout, prior to full production, where you only want to intercept a subset of the clients. After you are in full production mode, you can disable the Restricted Intercept List.

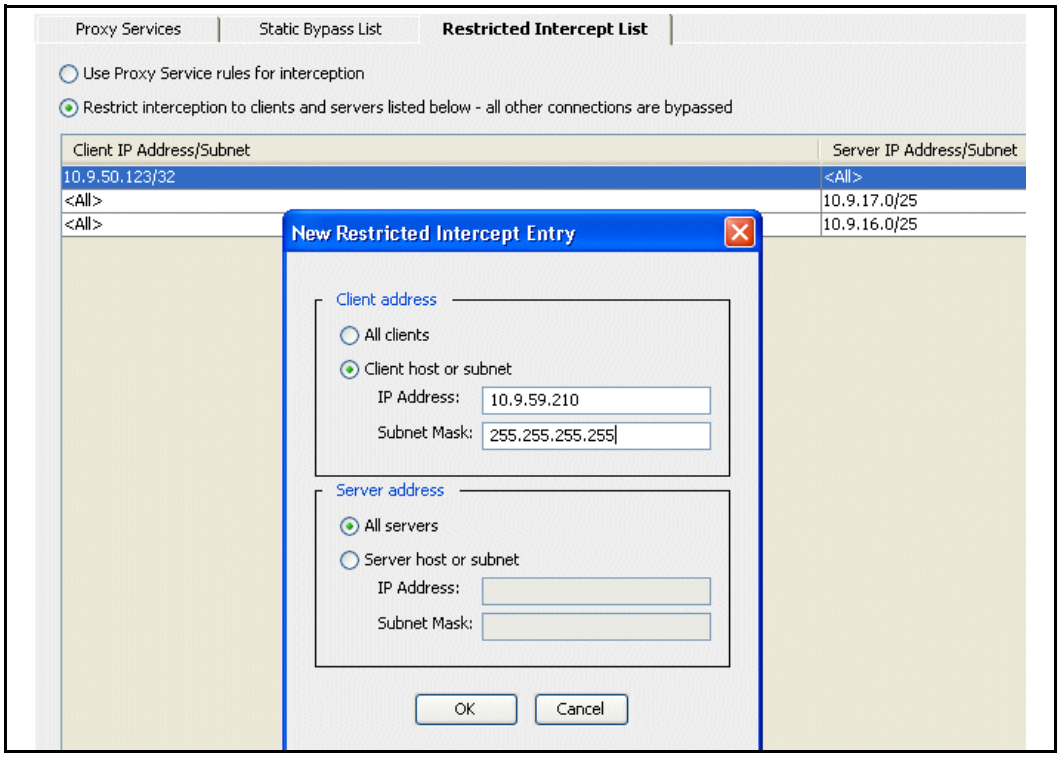
The Restricted Intercept List is also useful when troubleshooting an issue, because you can reduce the set of systems that are intercepted. If the restrict interception radio button (**Configuration > Services > Proxy Services > Restricted Intercept List**) is selected, any systems not on the list are bypassed.

If restricted intercept is disabled, the traffic behavior reverts to the previous behavior (before the Restricted Intercept List was enabled). If restricted intercept is enabled, traffic not in the list of systems is bypassed.

Note: An entry can exist in both the Static Bypass List and the Restricted Intercept List. However, the Static Bypass List overrides the entries in the Restricted Intercept List.

To configure a Restricted Intercept List:

1. Click **Configuration > Services > Proxy Services > Restricted Intercept List.**
2. Click **Restrict Interception to the servers and clients listed below-- all other connections are bypassed.**
3. Click **New** to create a new list entry, or click **Edit** to modify a list entry.



4. To select a specific client to be intercepted, click **Client host or subnet** and enter the **IP Address** and **Subnet Mask**. To select all clients using a specific server, click **All clients**, then enter the server **IP Address** and **Subnet Mask** in the Server address section.
5. Click **OK** to close the dialog.
6. Click **Apply**.

Related CLI Syntax to Configure Restricted Intercept Lists

- To enter configuration mode for the service:

```
SGOS#(config) proxy-services
SGOS#(config proxy-services) restricted-intercept
```

- The following subcommands are available:

```
SGOS#(config restricted-intercept) {enable | disable}
SGOS#(config restricted-intercept) add {all | client_ip | client_ip/
subnet-mask} | {all| server_ip | server_ip/subnet-mask}
SGOS#(config restricted-intercept) remove {all | client_ip |
client_ip/subnet-mask} | all| server_ip | server_ip/subnet-mask}
SGOS#(config restricted-intercept) view {<Enter> | filter {all |
client_ip | client_ip/subnet-mask} | {all| server_ip | server_ip/
subnet-mask}
```

Section F: Proxy Services and Listeners

Defaults:

- Proxy Edition: Table 3–5, "Proxy Name and Listeners" on page 55 lists the default ProxySG services and their default listeners. If you have an upgraded appliance, all services existing before the upgrade are preserved.
- MACH5 Edition:
  - A transparent TCP tunnel connection listening on port 23 is created in place of the default Telnet service.
  - Instant messaging, HTTPS reverse proxy, SOCKS, and Telnet services are not created on the MACH5 Edition ProxySG and are not included in trend data.

Note: Console services, used to manage the ProxySG, are not discussed in this chapter. For information about the four console services—HTTP, HTTPS, SSH, and Telnet—see Chapter 2: "About Management Services" on page 16.

Table 3–5 Proxy Name and Listeners

Service Name	Proxy	Destination IP Address	Port Range	Configuration Discussed
AOL-IM	AOL-IM	<All>	5190	*Volume 2: Proxies and Proxy Services*
CIFS	CIFS	<Transparent>	445, 139	Chapter 4: "Accelerating File Sharing" on page 59
Citrix ICA	TCP-Tunnel	<Transparent>	1494	Chapter 13: "Managing the TCP Tunneling Proxy" on page 207
DNS	DNS	<All>	53	Chapter 5: "Managing the Domain Name Service (DNS) Proxy" on page 79
Endpoint Mapper	Endpoint Mapper	<All>	135	Chapter 6: "Accelerating the Microsoft Outlook Application (Endpoint Mapper and MAPI Proxies)" on page 83
FTP	FTP	<All>	21	Chapter 7: "Managing the File Transport Protocol (FTP) Proxy" on page 101
HTTP	HTTP	<All>	80	Chapter 8: "Intercepting and Optimizing HTTP Traffic" on page 111
		<Explicit>	8080	
HTTPS	SSL	<All>	443	Chapter 12: "Managing the SSL Proxy" on page 177

Table 3–5 Proxy Name and Listeners (Continued)

Service Name	Proxy	Destination IP Address	Port Range	Configuration Discussed
IMAP	TCP-Tunnel	<Transparent>	143	Chapter 13: "Managing the TCP Tunneling Proxy" on page 207
IMAPS	TCP-Tunnel	<Transparent>	993	Chapter 13: "Managing the TCP Tunneling Proxy" on page 207
Kerberos	TCP-Tunnel	<Transparent>	88	Chapter 13: "Managing the TCP Tunneling Proxy" on page 207
LDAP	TCP-Tunnel	<Transparent>	389	Chapter 13: "Managing the TCP Tunneling Proxy" on page 207
LPD	TCP-Tunnel	<Transparent>	515	Chapter 13: "Managing the TCP Tunneling Proxy" on page 207
Lotus Notes	TCP-Tunnel	<Transparent>	1352	Chapter 13: "Managing the TCP Tunneling Proxy" on page 207
MMS	MMS	<All>	1755	*Volume 3: Web Communication Proxies*
MS SQL Server	TCP-Tunnel	<Transparent>	1433	Chapter 13: "Managing the TCP Tunneling Proxy" on page 207
MS Terminal Services	TCP-Tunnel	<Transparent>	3389	Chapter 13: "Managing the TCP Tunneling Proxy" on page 207
MSN-IM	MSN-IM	<All>	1863, 6891	*Volume 3: Web Communication Proxies*
MySQL	TCP-Tunnel	<Transparent>	3306	Chapter 13: "Managing the TCP Tunneling Proxy" on page 207
NFS	TCP-Tunnel	<Transparent>	2049	Chapter 13: "Managing the TCP Tunneling Proxy" on page 207
Novell GroupWise	TCP-Tunnel	<Transparent>	1677	Chapter 13: "Managing the TCP Tunneling Proxy" on page 207
Novell NCP	TCP-Tunnel	<Transparent>	524	Chapter 13: "Managing the TCP Tunneling Proxy" on page 207
Oracle	TCP-Tunnel	<Transparent>	1521, 1525	Chapter 13: "Managing the TCP Tunneling Proxy" on page 207
POP3	TCP-Tunnel	<Transparent>	110	Chapter 13: "Managing the TCP Tunneling Proxy" on page 207
POP3S	TCP-Tunnel	<Transparent>	995	Chapter 13: "Managing the TCP Tunneling Proxy" on page 207
RTSP	RTSP	<All>	554	*Volume 3: Web Communication Proxies*

Table 3–5 Proxy Name and Listeners (Continued)

Service Name	Proxy	Destination IP Address	Port Range	Configuration Discussed
Shell	TCP-Tunnel	<Transparent>	514	Chapter 10: "Managing Shell Proxies" on page 159
SMTP	TCP-Tunnel	<Transparent>	25	Chapter 13: "Managing the TCP Tunneling Proxy" on page 207
SOCKS		<Explicit>	1080	Chapter 11: "Managing a SOCKS Proxy" on page 167
SSH	TCP-Tunnel	<Transparent>	22	Chapter 13: "Managing the TCP Tunneling Proxy" on page 207
Sybase SQL	TCP-Tunnel	<Transparent>	1498	Chapter 13: "Managing the TCP Tunneling Proxy" on page 207
Telnet	Telnet	<All>	23	Chapter 10: "Managing Shell Proxies" on page 159
VNC	TCP-Tunnel	<Transparent>	5900	Chapter 13: "Managing the TCP Tunneling Proxy" on page 207
XWindows	TCP-Tunnel	<Transparent>	6000-6002	Chapter 13: "Managing the TCP Tunneling Proxy" on page 207
Yahoo-IM	Yahoo-IM	<All>	5050, 5101	*Volume 3: Web Communication Proxies*
Default (Listens on all unattended ports)	TCP-Tunnel	<Transparent>	<All>	Chapter 13: "Managing the TCP Tunneling Proxy" on page 207

Reference: Access Log Fields

The access log has two fields: service name and service group name.

- Name of the service used to intercept this connection:
  - `x-service-name` (ELFF token) `service.name` (CPL token)

Note: The `x-service-name` field replaces the `s-sitename` field. The `s-sitename` field can still be used for backward compatibility with squid log formats, but it has no CPL equivalent.

- Service group name:
  - `x-service-group` (ELFF token) `service.group` (CPL token)

Note: See *Volume 8: Access Logging*, Chapter 2 and *Access Log Formats*, Appendix B for detailed information about creating and editing log formats.

Reference: VPM Objects

The Service Group object can be configured in the Service column of the VPM Web Access layer.

For detailed information about VPM policy configuration, see *Volume 6: The Visual Policy Manager and Advanced Policy Tasks*, Chapter 3.

Reference: CPL Policy Configuration for Service Group

The following CPL is implemented per service group:

```
<proxy>
    service.group=standard reflect_ip(client)
```

The meaning of `reflect_ip(auto)` has changed in SGOS version 5.3 from deriving the reflect client ip setting from the service attribute to inheriting the reflect client ip setting from the global setting.

```
<proxy>
    service.group=interactive client.connection.dscp(preserve)
    service.group=interactive server.connection.dscp(preserve)
```

For detailed information about CPL policy configuration and revocation check, see *Volume 10: Content Language Policy Guide*, Chapter 4.

Chapter 4: Accelerating File Sharing

This chapter discusses file sharing optimization. File sharing uses the Common Internet File System (CIFS) protocol.

Topics in this Chapter

This chapter includes information about the following topics:

- "About the CIFS Protocol" on page 59
- "About the Blue Coat CIFS Proxy Solution" on page 60
- "Configuring the ProxySG CIFS Proxy" on page 62
- "Reference: Equivalent CIFS Proxy CLI Commands" on page 72
- "Reference: Access Log Fields" on page 73
- "Reference: CPL Triggers, Properties, and Actions" on page 76

About the CIFS Protocol

The CIFS protocol is based on the Server Message Block (SMB) protocol used for file sharing, printers, serial ports, and other communications. It is a client-server, request-response protocol. The CIFS protocol allows computers to share files and printers, supports authentication, and is popular in enterprises because it supports all Microsoft operating systems, clients, and servers.

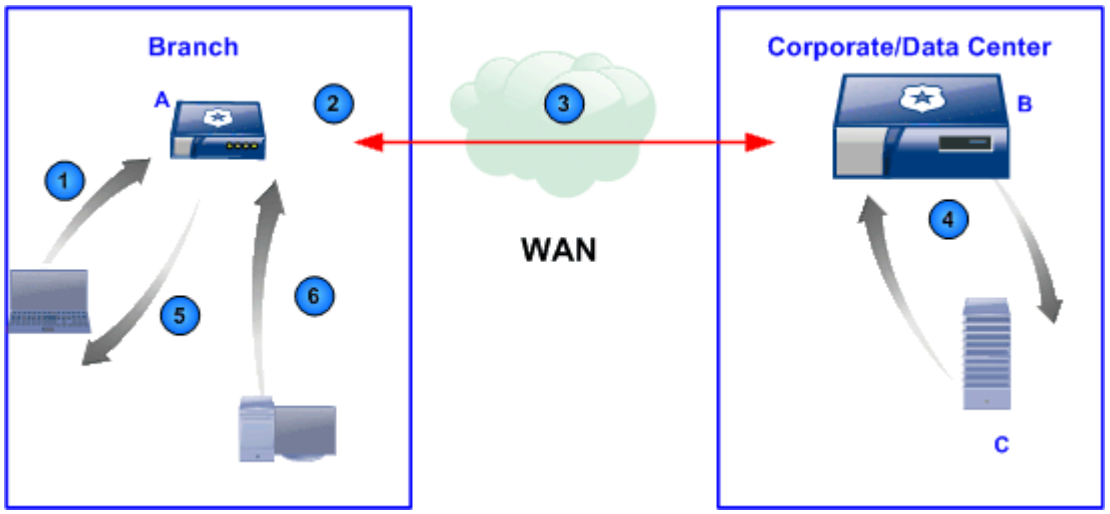
File servers make file systems and other resources (printers, mailslots, named pipes, APIs) available to clients on the network. Clients have their own hard disks, but they can also access shared file systems and printers on the servers.

Clients connect to servers using TCP/IP. After establishing a connection, clients can send commands (SMBs) to the server that allows them to access shares, open files, read and write files— the same tasks as with any file system, but over the network.

CIFS is beneficial because it is generic and compatible with the way applications already share data on local disks and file servers. More than one client can access and update the same file, while not compromising file-sharing and locking schemes. However, the challenge for an enterprise is that CIFS communications are inefficient over low bandwidth lines or lines with high latency, such as in enterprise branch offices. This is because CIFS transmissions are broken into *blocks* of data (typically close to 64 KB). The client must stop and wait for each block to arrive before requesting the next block. Each stop represents time lost instead of data sent. Therefore, users attempting to access, move, or modify documents experience substantial, work-prohibiting delays.

About the Blue Coat CIFS Proxy Solution

The CIFS proxy on the ProxySG combines the benefits of the CIFS protocol with the abilities of the ProxySG to improve performance, reduce bandwidth, and apply basic policy checks. This solution is designed for branch office deployments because network administrators can consolidate their Windows file servers (at the core office) instead of spreading them across the network.



LEGEND:

A: Branch ProxySG.

B: Concentrator ProxySG.

C: File server containing objects requested by branch users.

DATA FLOW:

1: A branch client requests a file from a server at the core office.

2: If the branch proxy has the object or part of the object cached, it is served back to the client; otherwise, the request for uncached objects is sent to the core office. With the default CIFS proxy configuration, the ProxySG attempts to read ahead—anticipate what part(s) of a specific object might be requested next.

3: With the default CIFS proxy configuration, byte caching compresses the data over the TCP connection.

4: The Concentrator performs decompression and authentication tasks, accesses the content server, and returns the content back to the branch.

5. The client receives the requested content. In addition, the anticipated content is cached so that future requests for it can be served without requesting it from the data center.

6. Another client requests access to a file on the core server, but wants to write to the file. With the default CIFS proxy configuration, the branch ProxySG continuously informs the client that has written the data and that it is okay to write the next block. Simultaneously, the ProxySG sends the data over the WAN to the file server, thus maximizing the data pipeline.

Figure 4–1 CIFS Proxy Traffic and Flow Diagram

Caching Behavior

The CIFS proxy caches the regions of files that are read or written by the client (partial caching) and applies to both read and write file activities. Also, the caching process respects file locking.

Note: Caching behavior can also be controlled with policy.

Authentication

The CIFS proxy supports both server and proxy authentication in the following contexts.

Server Authentication

Permissions set by the origin content server (OCS) are always honored. Requests to open a file are forwarded to the OCS; if the OCS rejects the client access request, no content is served from the cache.

Note: NTLM/IWA authentication requires that the client knows what origin server it is connecting to so it can obtain the proper credentials from the domain controller.

Proxy Authentication

The ProxySG cannot issue a challenge to the user over CIFS, but it is able to make use of credentials acquired by other protocols if IP surrogates are enabled.

Policy Support

The CIFS proxy supports the `proxy`, `cache`, and `exception` policy layers. However, the SMB protocol can only return error numbers. Exception definitions in the forms of strings cannot be seen by an end user. See "Reference: CPL Triggers, Properties, and Actions" on page 76 for supported CPL triggers and actions.

Access Logging

By default, the ProxySG uses a Blue Coat-derived CIFS access log format.

```
date time c-ip r-ip r-port x-cifs-method x-cifs-server x-cifs-share
x-cifs-path x-cifs-orig-path x-cifs-client-bytes-read
x-cifs-server-bytes-read x-cifs-bytes-written x-cifs-file-type
s-action cs-username cs-auth-group s-ip
```

For a reference list and descriptions of used log fields, see "Reference: Access Log Fields" on page 73.

WCCP Support

If WCCP is deployed for transparency, you must configure WCCP to intercept TCP ports 139 and 445.

Configuring the ProxySG CIFS Proxy

This section contains the following sub-sections:

- "About Windows Security Signatures" on page 62
- "Intercepting CIFS Services" on page 63
- "Configuring the CIFS Proxy Options" on page 67
- "Reviewing CIFS Protocol Statistics" on page 69

About Windows Security Signatures

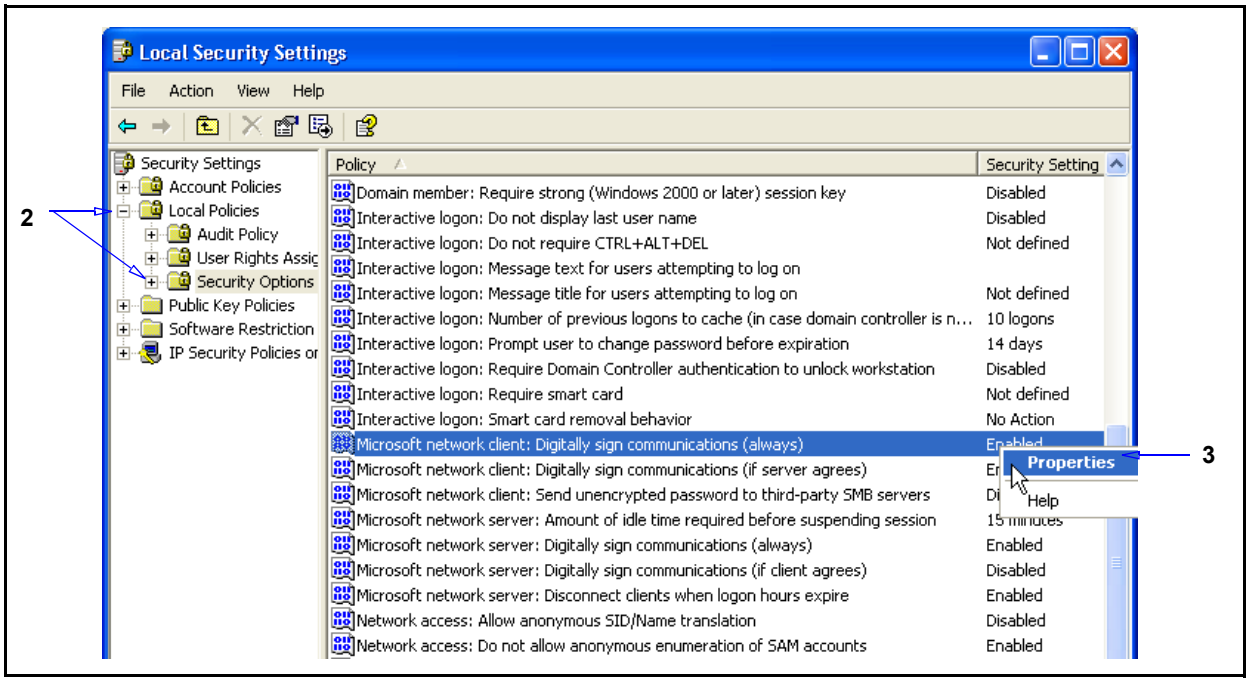
Security signatures prevent the CIFS proxy from providing its full acceleration capabilities. Additionally, security signatures require a considerable amount of processing on both clients and servers. As their benefits are often superseded by link-layer security measures, such as VPNs and restricted network topology, the benefits are minimal and the drawbacks are high. The CIFS proxy requires that security signatures are disabled.

If you know this setting is disabled on your clients or servers, you can proceed to "Configuring the CIFS Proxy Options" on page 67.

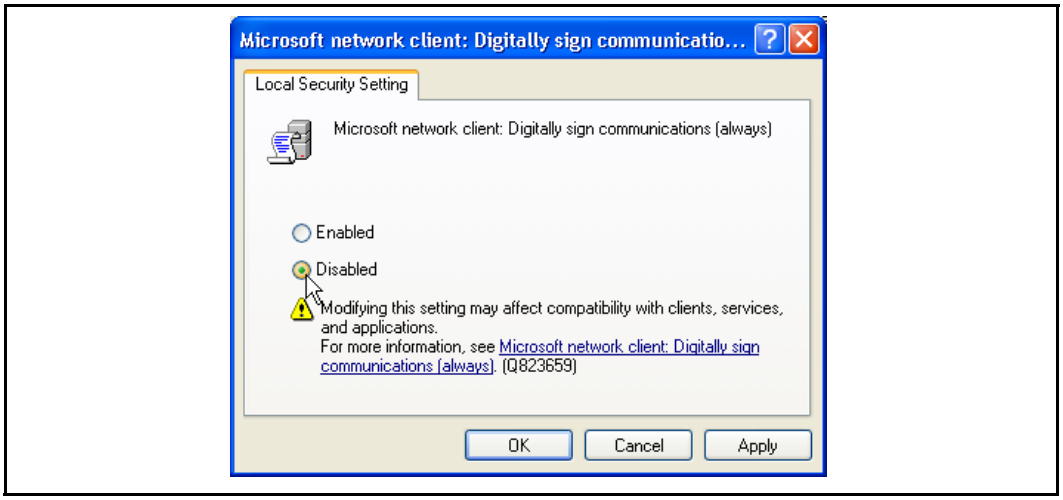
To verify the state of security signatures in Windows; disable if necessary:

Note: This procedure follows the Control Panel Classic View format. The screen shots represent Microsoft Windows XP.

1. In Windows, select **Start > Control Panel > Administrative Tools > Local Security Policy**. The Local Security Settings dialog appears.



2. Select **Local Policies > Security Options**.
3. Perform one of the following:
   - Windows XP / 2003: Right-click **Microsoft network client: Digitally sign communications (always)** and select **Properties**. A configuration dialog appears.
   - Windows 2000: Right-click **Digitally sign client communications (always)**. A configuration dialog appears.



4. Select **Disabled**. Click **Apply** and **OK**.
5. Repeat for the server options:
   - Windows XP / 2003: Right-click **Microsoft network server: Digitialy sign communications (always).**
   - Windows 2000: Right-click **Digitally sign server communications (always).**
6. Close all Control Panel dialogs.

Important: If the server is an ADS / Domain controller, you must set the same security settings for both **Administrative Tools > Domain Controller Security Policy** and **Administrative Tools- > Domain Security Policy.** Otherwise, you cannot open file shares and Group Policy snap-ins on your server.

---

7. You must reboot the client or server to apply this configuration change.

Intercepting CIFS Services

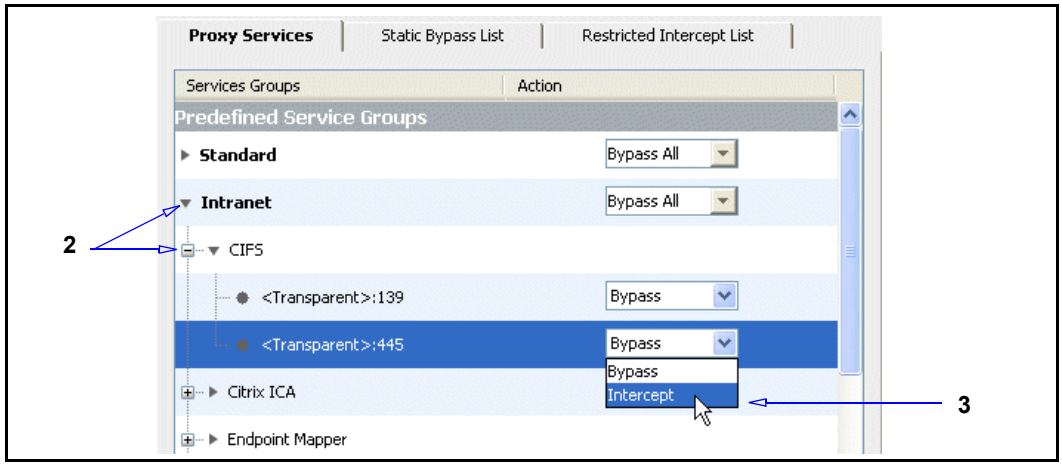
By default (upon upgrade and on new systems), the ProxySG has CIFS services configured for transparent connections on ports 139 and 445. Blue Coat creates listener services on both ports because different Windows operating systems (older versus newer) attempt to connect using 139 or 445. For example, Windows NT and earlier only used 139, but Windows 2000 and later try both 139 and 445. Therefore only configuring one port can potentially cause only a portion of Windows 2000 and newer CIFS traffic to go through the proxy.

A transparent connection is the only supported method; the CIFS protocol does not support explicit connections.

Also, by default these services are configured to accept all IP addresses in **Bypass** mode. The following procedure describes how to change them to **Intercept** mode, and explains other attributes within the service.

To configure the CIFS proxy to intercept file sharing traffic:

1. From the Management Console, select **Configuration > Services > Proxy Services**.



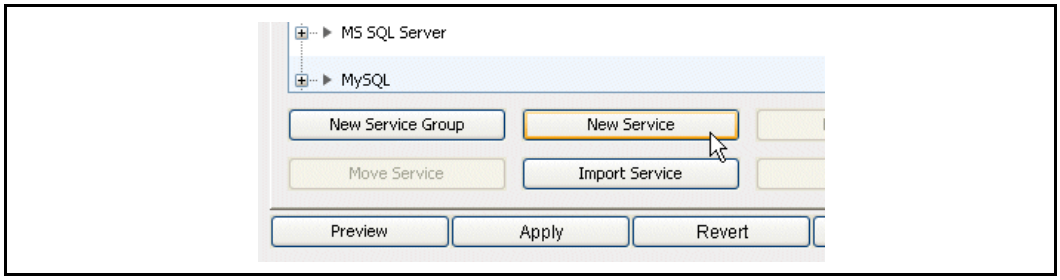
2. Scroll the list of service groupgs, click **Intranet**, and click **CIFS** to expand the CIFS services list.
3. Notice the **Action** for each default service (ports 139 and 445) is **Bypass**. Select **Intercept** from the drop-down list(s).
4. Click **Apply**.

Adding and Configuring New CIFS Services

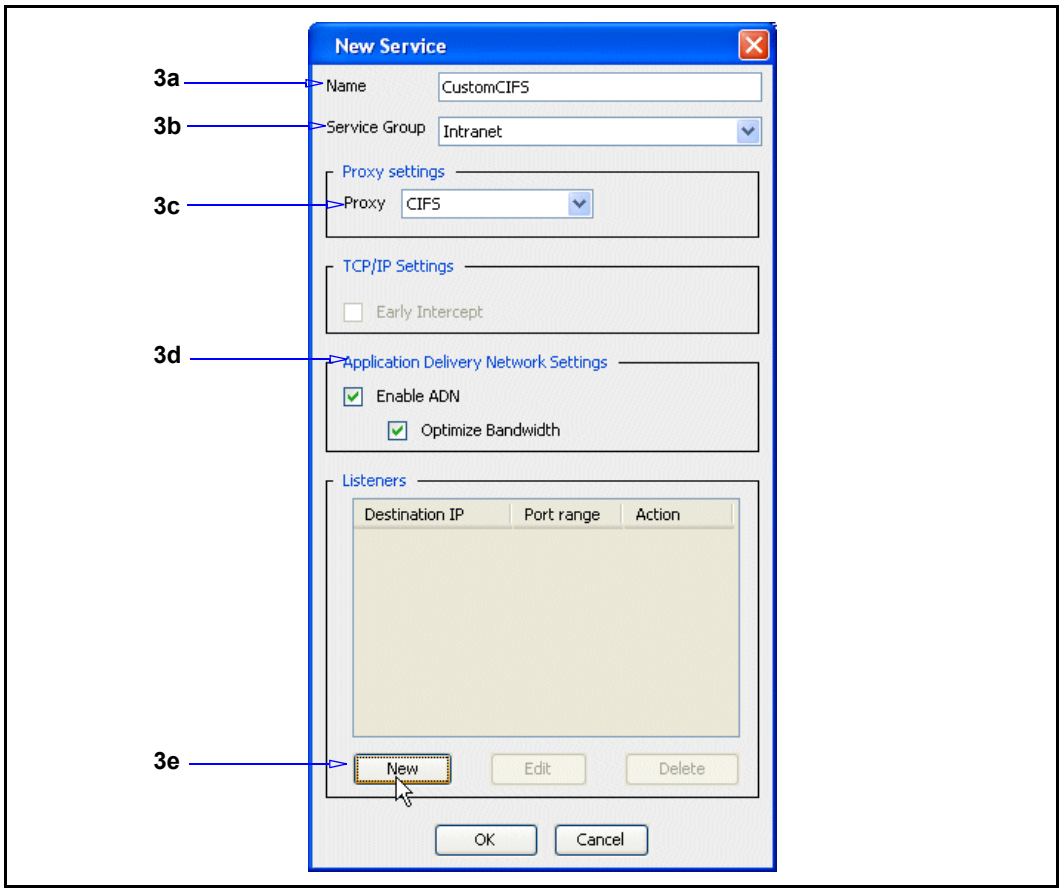
If you require a CIFS service to intercept a port other than the default `139/445` ports, you can create a new service.

To create and configure a new CIFS service:

1. From the Management Console, select **Configuration > Services > Proxy Services**.



2. At the bottom of the page, click **New Service**. The New Service dialog displays.



3. Configure the service attributes:

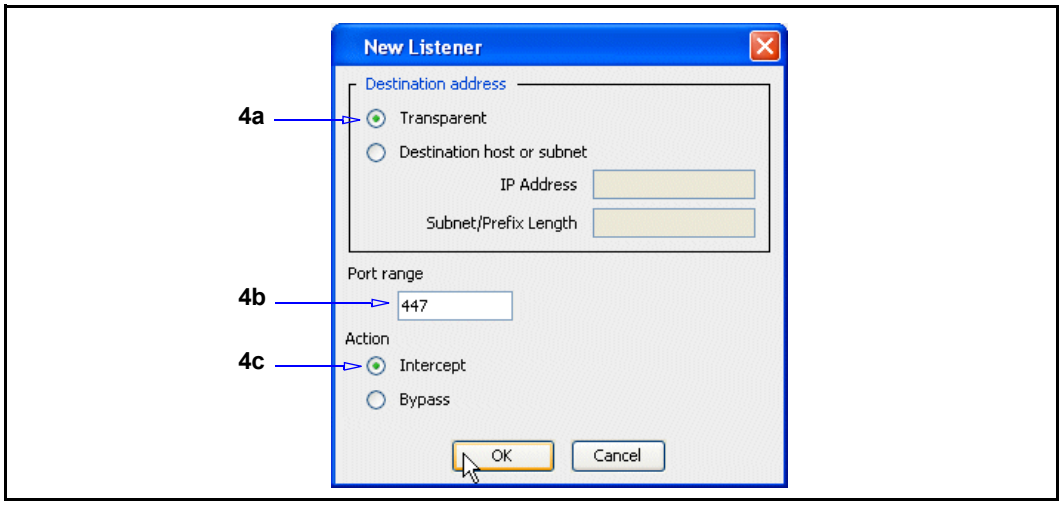
   a. Name the service (tip: cannot use the default name **CIFS**).

   b. From the **Service Group** drop-down list, select **Intranet** (for all users except mobile users (ProxyClient), CIFS traffic occurs over the enterprise Intranet).

   c. From the **Proxy Settings** drop-down list, select **CIFS**.

   d. Blue Coat recommends selecting the **Enable ADN** and Optimize Bandwidth options. These feature improve performance by compressing request and response data, which still needs to be forwarded across the WAN. For more information about ADN optimization, refer to *Volume 5: Advanced Networking*.

   e. In the **Listeners** area, click **New**. The New Listener dialog displays.



4. Configure the new listener attributes:
   a. Select **Transparent**.
   b. Enter a port number (range, such as `347-350`).
   c. Select **Intercept**.
   d. Click **OK** to close the dialog.
5. Click **OK** to close the New Service dialog.

Result: The CIFS service is configured and displays in the list under the Intranet service group.

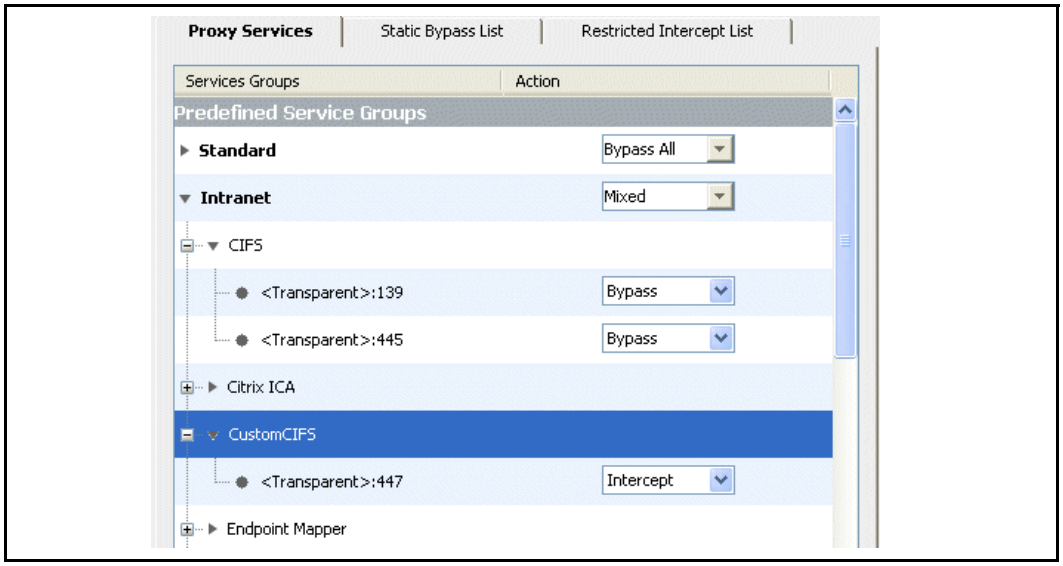


Figure 4–2 Custom CIFS service added to the Intranet service group.

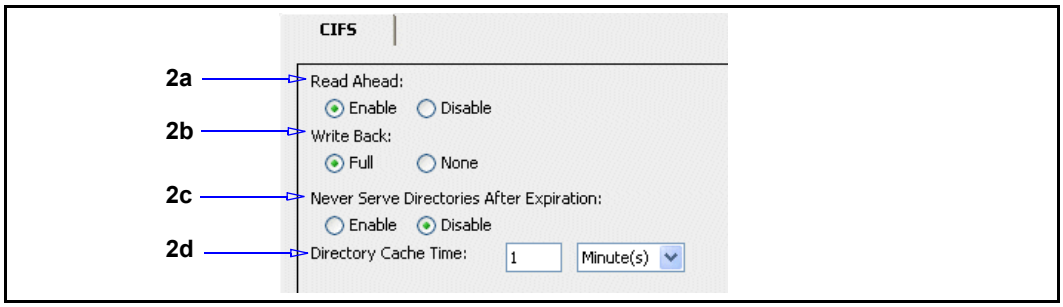
Now that the CIFS listeners are configured, you can configure the CIFS proxy.

Configuring the CIFS Proxy Options

The CIFS proxy options configure file reading and writing and folder management. These options are enabled by default because they maximize the benefits of a CIFS proxy deployment. This section describes these options and why they might require changing based on your branch deployment.

To view/change t he CIFS proxy configuration options:

1. In the Management Console, select **Configuration > Proxy Settings > CIFS Proxy.**



2. Configure the CIFS proxy options:

   a. **Read ahead**: Enabled by default, which reduces the latency of the connection. The ProxySG might partially cache a requested object (the part directly requested and viewed by the client). When **Read ahead** is enabled, the appliance attempts to *anticipate* what data might be requested next, fetches it, and caches it.

      If applications are performing a large amount of non-sequential file access, disabling **Read Ahead** reduces the amount of unnecessary data being fetched into the cache.

   b. **Write back**: Enabled by default. This option applies to when clients attempt to write to a file on the core server. Without the CIFS proxy, a client would experience substantial latency as it sends data chunks and waits for the acknowledgement from the server to write the next data chunks. With this option enabled the branch ProxySG is viewed by the client as the file server; the appliance constantly sends approval to the client and allows the client to write data while on the back end takes advantage of the compressed TCP connection and sends the data to the core server.

      A reason for disabling this option is the risk of data loss if the link from the branch to the core fails. There is no way to recover queued data if such a link failure occurs.

   c. **Never Serve Directories After Expiration:** Disabled by default. When this option is enabled and **Directory Cache Time** has a value of 0, directories are refreshed synchronously instead of in the background. This is needed when the set of visible objects in a directory returned by a server can vary between users.

   d. **Directory Cache Time**: This option determines how long directory information is kept in cache. Changes made to a directory by clients not using the ProxySG are not visible to ProxySG clients if they occur within this time interval. The default cache time is 30 seconds. Blue Coat recommends keeping this value low to ensure clients have access

to the most current directory information; however, you can set it longer if your applications use CIFS to access files. For example, the cache responds faster if it knows directory X does not contain the file and so moves on to directory Y, which reduces the number of round trips to the file server.

3. If you changed any options, click **Apply**.

Enabling CIFS Access Logging

By default, the ProxySG is configured to use the Blue Coat CIFS access log format. Access Logging is enabled on the **Configuration > Access Logging > General** page.

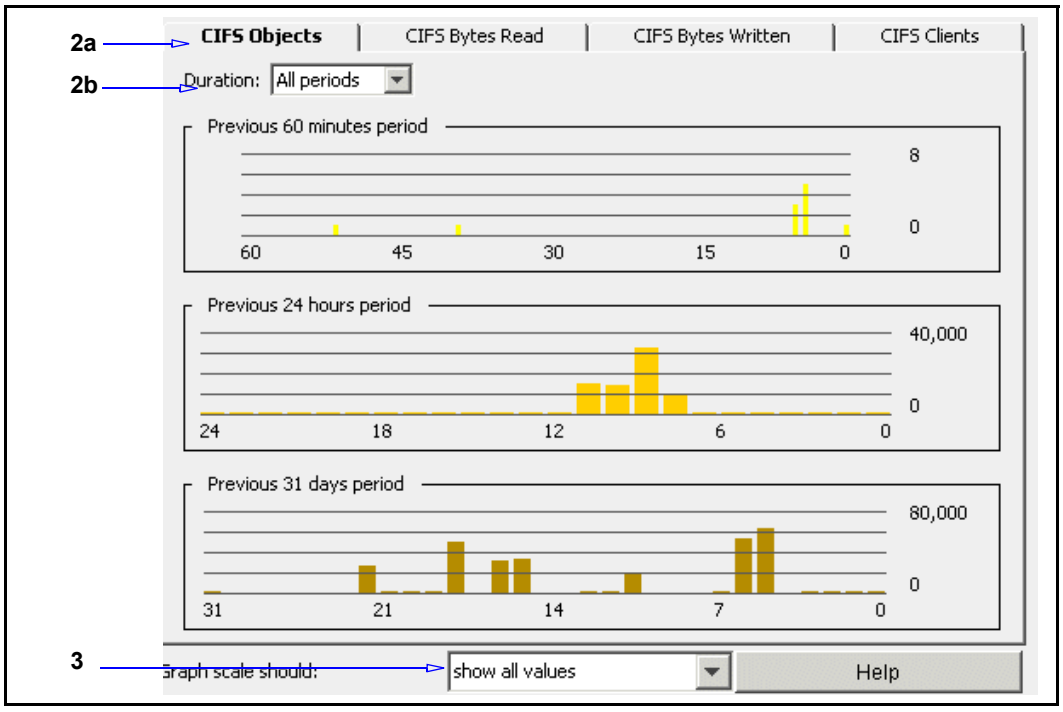
For information about access log customization, refer to *Volume 8: Access Logging*.

Reviewing CIFS Protocol Statistics

After CIFS traffic begins to flow through the ProxySG, you can review the statistics page and monitor results in various CIFS categories. The presented statistics are representative of the client perspective.

To review CIFS statistics:

1. From the Management Console, select **Statistics > Protocol Details > CIFS History.**



2. View statistics:
   a. From the **Service** or **Proxy** drop-down list, select **CIFS**.
   b. Select a statistic category tab:
      - **CIFS Objects:** The total number of CIFS-related objects processed by the ProxySG (read and written).

- **CIFS Bytes Read**: The total number of bytes read by CIFS clients.
- **CIFS Bytes Written**: The total number of bytes written by CIFS clients (such as updating existing files on servers).
- **CIFS Clients**: The total number of connected CIFS clients.
- **CIFS Bandwidth Gain**: The total bandwidth usage for clients (yellow) and servers (blue), plus the percentage gain.

c. The graphs display three time metrics: the previous 60 minutes, the previous 24 hours, and the previous 30 days. Select **Duration**: from the drop-down list. Roll the mouse over any colored bar to view details.

3. (Optional) You can change the scale of the graph to display the percentage of bar peaks to display.

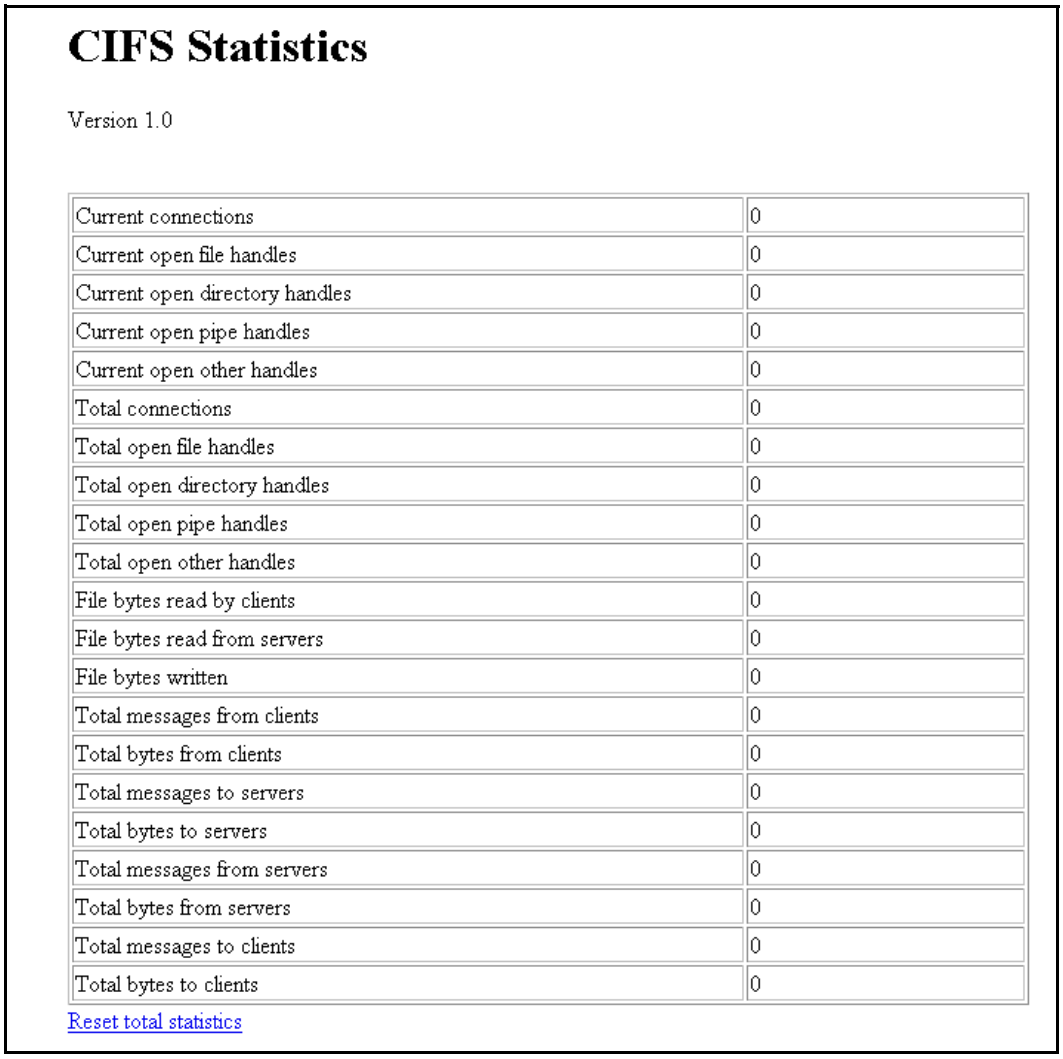
Statistic URL Pages

Additional CIFS statistics pages are viewable from Management Console URLs.

Statistics

This page displays various, more granular connection and byte statistics.

`https://SG_IP_address:8082/CIFS/statistics`



CIFS Statistics

Version 1.0

Current connections	0
Current open file handles	0
Current open directory handles	0
Current open pipe handles	0
Current open other handles	0
Total connections	0
Total open file handles	0
Total open directory handles	0
Total open pipe handles	0
Total open other handles	0
File bytes read by clients	0
File bytes read from servers	0
File bytes written	0
Total messages from clients	0
Total bytes from clients	0
Total messages to servers	0
Total bytes to servers	0
Total messages from servers	0
Total bytes from servers	0
Total messages to clients	0
Total bytes to clients	0

Reset total statistics

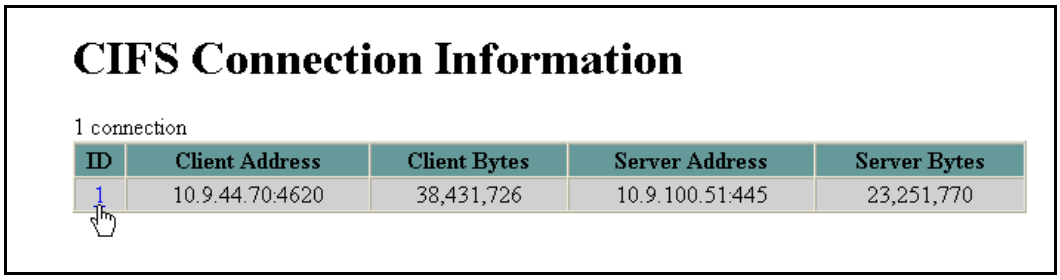
If CIFS traffic interception is occurring (the above screenshot does not represent active traffic), the byte counters increment when a user opens a file or browses around.

Note: The `bytes to/from servers` counters on the CIFS statistics page do *not* include the effects of compression and byte caching over the WAN link.

Connections

This page displays specific client-to-server connection and file information and statistics.

`https://ProxySG_`*`IP_address`*`:8082/CIFS/connection`

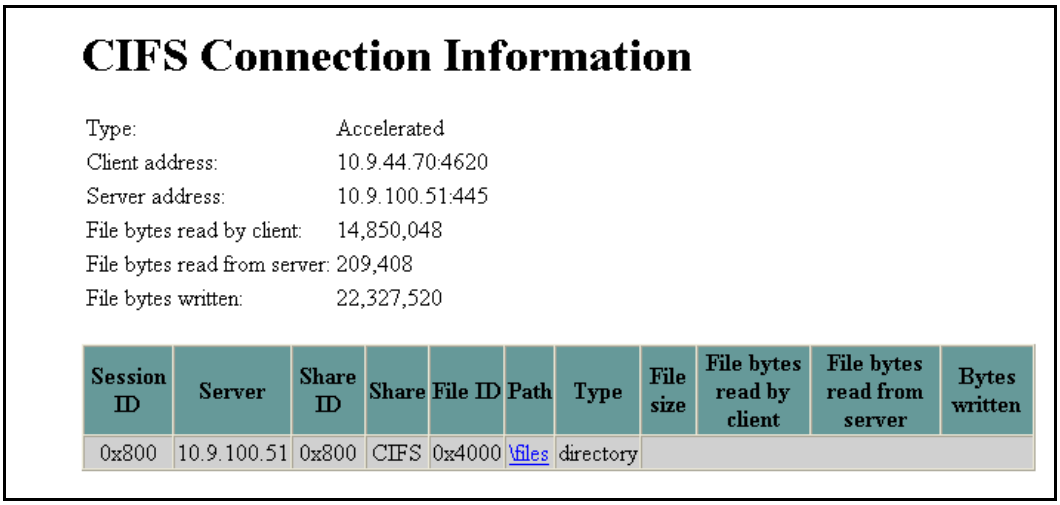


CIFS Connection Information

1 connection

ID	Client Address	Client Bytes	Server Address	Server Bytes
1	10.9.44.70:4620	38,431,726	10.9.100.51:445	23,251,770

Click **connection ID link** to drill down to more details.



CIFS Connection Information

Type: Accelerated
Client address: 10.9.44.70:4620
Server address: 10.9.100.51:445
File bytes read by client: 14,850,048
File bytes read from server: 209,408
File bytes written: 22,327,520

Session ID	Server	Share ID	Share	File ID	Path	Type	File size	File bytes read by client	File bytes read from server	Bytes written
0x800	10.9.100.51	0x800	CIFS	0x4000	\files	directory				

Reference: Equivalent CIFS Proxy CLI Commands

The Management Console procedures in this chapter have the following equivalent CLI command roots:

- To enter configuration mode for the service:

```
SGOS#(config) proxy-services
SGOS#(config proxy-services) create cifs service-name
SGOS#(config proxy-services) edit service-name
```

- The following subcommands are available:

```
SGOS#(config service-name) add {transparent | ip_address | ip_address/
subnet-mask} {port | first_port-last_port} [intercept | bypass]
SGOS#(config service-name) attribute {adn-optimize {enable | disable}
SGOS#(config service-name) bypass {transparent | ip_address |
ip_address/subnet-mask} {port | first_port-last_port}}
SGOS#(config service-name) exit
SGOS#(config service-name) intercept {transparent | ip_address |
ip_address/subnet-mask} {port | first_port-last_port}}
SGOS#(config service-name) remove {transparent | ip_address |
ip_address/subnet-mask} {port | first_port-last_port}
SGOS#(config service-name) view
```

- To set other configuration parameters:

```
SGOS#(config service-name) exit
SGOS#(config) cifs
SGOS#(config cifs)
```

- The following subcommands are available:

```
SGOS#(config cifs) directory-cache-time seconds
SGOS#(config cifs) exit
SGOS#(config cifs) read-ahead {disable | enable}
SGOS#(config cifs) strict-directory-expiration {disable | enable}
SGOS#(config cifs) view {configuration | statistics}
SGOS#(config cifs) write-back {full | none}
```

Reference: Access Log Fields

The default Blue Coat CIFS Access Log fields are:

- `c-ip`: IP address of the CIFS client.
- `c-port`: The CIFS client port TCP connection.
- `cs-auth-group`: One group that an authenticated user belongs to. If a user belongs to multiple groups, the group logged is determined by the Group Log Order configuration specified in VPM. If the Group Log Order is not specified, an arbitrary group is logged. Note that only groups referenced by policy are considered.
- `cs-username`: Relative username of a client authenticated to the proxy (for example: not fully distinguished).
- `r-ip`: IP address from the outbound server URL.
- `r-port`: Port from the outbound server URL, typically 139 or 445.
- `s-action`: The logging action (or flow) being one of the following:
  - `ALLOWED`: CIFS operation passed the policy checkpoint and was also successful.
  - `DENIED`: CIFS operation failed the policy checkpoint.
  - `ERROR`: CIFS operation resulted in an error on the server; typically associated with NT (`x-cifs-nt-error-code`) or DOS error (`x-cifs-dos-error-code`, `x-cifs-dos-error-class`).
  - `FAILED`: CIFS operation was successful on the server but failed on the proxy for some internal reason.
  - `SUCCESS`: CIFS operation was successful on the server (did not go through policy checkpoint).
- `s-ip`: IP address of the appliance on which the client established its connection.

- x-cifs-client-bytes-read: Total number of bytes read by a CIFS client from the associated resource. For **OPEN/CLOSE**, it is the total for that specific file. For **MOUNT/UNMOUNT**, the total for all files accessed in that share. For **LOGON/LOGOFF**, the total for all files accessed in that session. For **CONNECT/DISCONNECT**, the total for all files accessed during that connection.
- x-cifs-client-write-operations: Total number of client write operations for this particular resource. The scope is the same as x-cifs-client-read-operations.
- x-cifs-client-other-operations: Total number of client operations that are not reads or writes for this particular resource. The scope is the same as x-cifs-client-read-operations. **MOUNT/UNMOUNT** might also include operations not tied to a specific open file.
- x-cifs-bytes-written: Total number of bytes written to the associated resource.
- x-cifs-dos-error-class: DOS error class generated by server, in hexadecimal.
- x-cifs-dos-error-code: DOS error code generated by server, in hexadecimal.
- x-cifs-error-cod: CIFS error code generated by server. If the error code is in NT format, it is a single hexadecimal number of the form 0xNNNNNNNN. If the error code is in DOS format, it is two hexadecimal numbers of the form 0xNN/0xNNNN. The first number is the DOS error class, and the second is the DOS error code. This field is a combination of the x-cifs-nt-error-code, x-cifs-dos-error-class, and x-cifs-dos-error-code.
- x-cifs-fid: Numeric ID representing a CIFS resource.
- x-cifs-file-size: Size in bytes of CIFS resource.
- x-cifs-file-type: The type of file that was opened or closed. Values are file, directory, pipe, or other. It is only valid if x-cifs-method is OPEN, CLOSE, CLOSE_ON_UNMAP, CLOSE_ON_LOGOFF, CLOSE_ON_DISCONNECT, or CLOSE_ON_PASSTHRU.
- x-cifs-method: The method associated with the CIFS request. The list of CIFS methods are:
  - CONNECT: For TCP-level connect from client to CIFS server.
  - DISCONNECT: For TCP-level connection shutdown.
  - LOGON: For SESSION_SETUP_ANDX SMB command.
  - LOGOFF: For LOGOFF_ANDX SMB command.
  - LOGOFF_ON_PASSTHRU: For removal of cached session from proxy upon PASSTHRU.
  - LOGOFF_ON_DISCONNECT: For removal of cached session from proxy upon DISCONNECT.
  - MAP: For TREE_CONNECT SMB command.
  - UNMAP: For TREE_DISCONNECT SMB command.
  - UNMAP_ON_LOGOFF: For removal of cached share from proxy upon LOGOFF.

- UNMAP_ON_PASSTHRU: For removal of cached share from proxy upon PASSTHRU.
- UNMAP_ON_DISCONNECT: For removal of cached share from proxy upon DISCONNECT.
- DELETE: For path-based DELETE and DELETE_DIRECTORY SMB commands.
- DELETE_ON_CLOSE: For delete-on-close action done on a CIFS resource.
- LIST: For enumerating contents of a directory.
- OPEN: For opening a CIFS resource.
- RENAME: For renaming a CIFS resource.
- CLOSE: For closing a CIFS resource.
- CLOSE_ON_UNMAP: For removal of cached file from proxy upon UNMAP.
- CLOSE_ON_LOGOFF: For removal of cached file from proxy upon LOGOFF.
- CLOSE_ON_PASSTHRU: For removal of cached file from proxy upon PASSTHRU.
- CLOSE_ON_DISCONNECT: For removal of cached file from proxy upon DISCONNECT.
- PASSTHRU: For connections which Blue Coat is unable to handle:
  - Client or server does not support NTLM 0.12 dialect.
  - Security signatures are enabled.
  - Client or server does not support Unicode characters.
  - The SESSION_SETUP_ANDX SMB request is malformed (with unknown word count).
  - Header portion of some SMB command is malformed.
  - NETBIOS header is malformed.
- OPEN_STATS: Log the same fields as CLOSE for gathering time-based activity information on open files. This occurs on a 5 minute interval if there was activity on the file within that interval.

- x-cifs-nt-error-code: CIFS error code generated by server, in hexadecimal.
- x-cifs-orig-path: Original path name of resource to be renamed.
- x-cifs-orig-unc-path: UNC path of original path name of resource to be renamed.
- x-cifs-path: CIFS resource name as specified in the UNC path.
- x-cifs-server: CIFS server as specified in the UNC path.
- x-cifs-server-bytes-read: Total number of bytes read from CIFS server from the associated resource.
- x-cifs-server-operations: Total number of server operations for this particular resource. The scope is the same as x-cifs-client-read-operations.
- x-cifs-share: CIFS share name as specified in the UNC path.

- `x-cifs-tid`: ID representing instance of an authenticated connection to server resource.
- `x-cifs-uid`: ID representing an authenticated user instance.
- `x-cifs-unc-path`: CIFS path of form `\\server\share\`*`path`* where *`path`* might be empty.
- `x-client-connection-bytes`: Number of bytes sent to and received from the client.
- `x-server-connection-bytes`: Number of bytes sent to and received from the server. If ADN is used for the server connection, this is the number of bytes before ADN compression is applied.
- `x-server-adn-connection-bytes`: Number of bytes sent to and received from the server-side ADN peer if ADN is used for the server connection. If ADN is not used, this is displayed as "-".

Reference: CPL Triggers, Properties, and Actions

The following CPL applies to CIFS policy:

Triggers

- `attribute.<name>=, has_attribute.<name>=`
- `client.address=, client.host=, client.host.has_name=`
- `client.protocol=cifs`
- `content_management=no`
- `condition=`
- `date[.utc]=, day=, hour=, minute=, month=, weekday=, year=, time=`
- `has_client=`
- `proxy.address=, proxy.port=, proxy.card=`
- `raw_url=`
- `release.*=`
- `server_url=`
- `service.name=cifs`
- `tunneled=`
- `url=cifs://<ip>:<port>/`
- `user.*=, group=, realm=, authenticated=`

Properties and Actions:

- `action()`
- `access_log.*(), log.*(), log_message(), notify_email(), notify_snmp()`
- `adn.server.optimize(yes|no)`
- `adn.server.optimize(byte_cache)`
- `adn.server.optimize(compress)`

- `adn.server.optimize.inbound(yes|no)`
- `adn.server.optimize.outbound(yes|no)`
- `adn.connection.dscp(DSCP_value | DSCP_name | preserve)`
- `authenticate.*()`
- `allow, deny, deny.*(), exception.*(), force_deny.*(), force_exception.*()`
- `bypass_cache()`
- `detect_protocol(cifs), force_protocol(cifs)`
- `limit_bandwidth(bandwidth_class)`
- `reflect_ip()`
- `rewrite(url), rewrite(url.host), set(url.port)`
- `trace.*()`

Chapter 5: Managing the Domain Name Service (DNS) Proxy

This chapter discusses managing Domain Name Service (DNS) traffic through the DNS proxy on the ProxySG (to configure the ProxySG connections to DNS servers, see *Volume 1: Getting Started*). When a DNS proxy service is enabled, it listens on port 53 for both explicit and transparent DNS domain query requests. By default, the service is created but not enabled.

The DNS performs a lookup of the DNS cache to determine if requests can be answered. If yes, the ProxySG responds. If not, the DNS forwards the request to the DNS server list configured on the on the ProxySG**.** (To configure the DNS server list, see **Configuration > Network > DNS**.)

Note: The ProxySG is not a DNS server. It does not perform zone transfers, and recursive queries are forwarded to other name servers.

For information on managing DNS name servers, refer to *Volume 1: Getting Started*.

Through policy, you can configure the list of resolved domain names (the *resolving name list*) the DNS uses. The domain name in each query received by the ProxySG is compared against the resolving name list. Upon a match, the appliance checks the resolving list. If a domain name match is found but no IP address was configured for the domain, the appliance sends a DNS query response containing its own IP address. If a domain name match is found with a corresponding IP address, that IP address is returned in a DNS query response. All unmatched queries are sent to the name servers configured on the ProxySG.

Topics in this Chapter

This chapter includes information about the following topics:

- "Creating or Editing a DNS Proxy Service" on page 80
- "Creating a Resolving Name List" on page 82

Configuring the DNS Proxy Service Options

This section describes how to change the default service options and add new services.

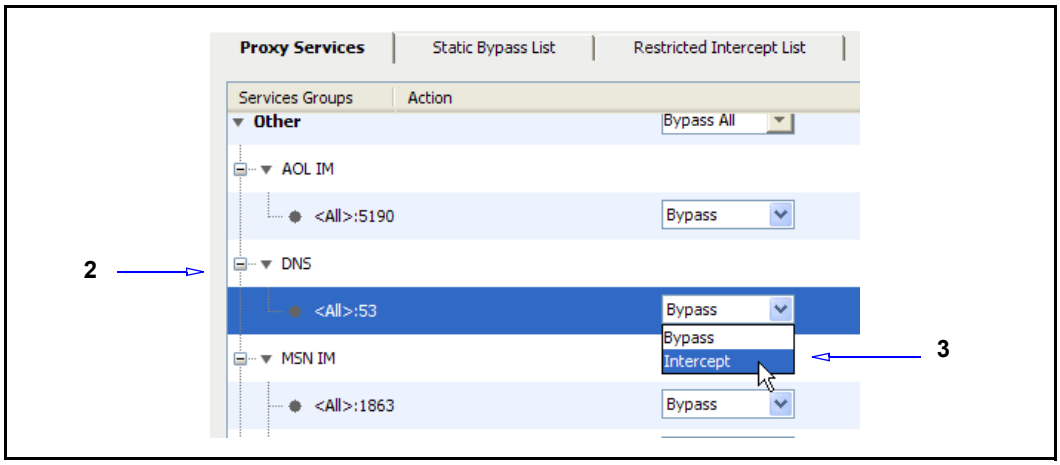
Changing the Default DNS Proxy Service to Intercept All IP Addresses on Port 53

By default (upon upgrade and on new systems), the ProxySG has an DNS proxy service configured on port 53. The service is configured to listen to all IP addresses, but is set in Bypass mode.

The following procedure describes how to change the service to Intercept mode.

To configure the DNS proxy to intercept traffic:

1. From the Management Console, select **Configuration > Services > Proxy Services.**
2. To edit an existing DNS proxy service, highlight the service and click **Edit.** To create a new proxy service, click **New.**

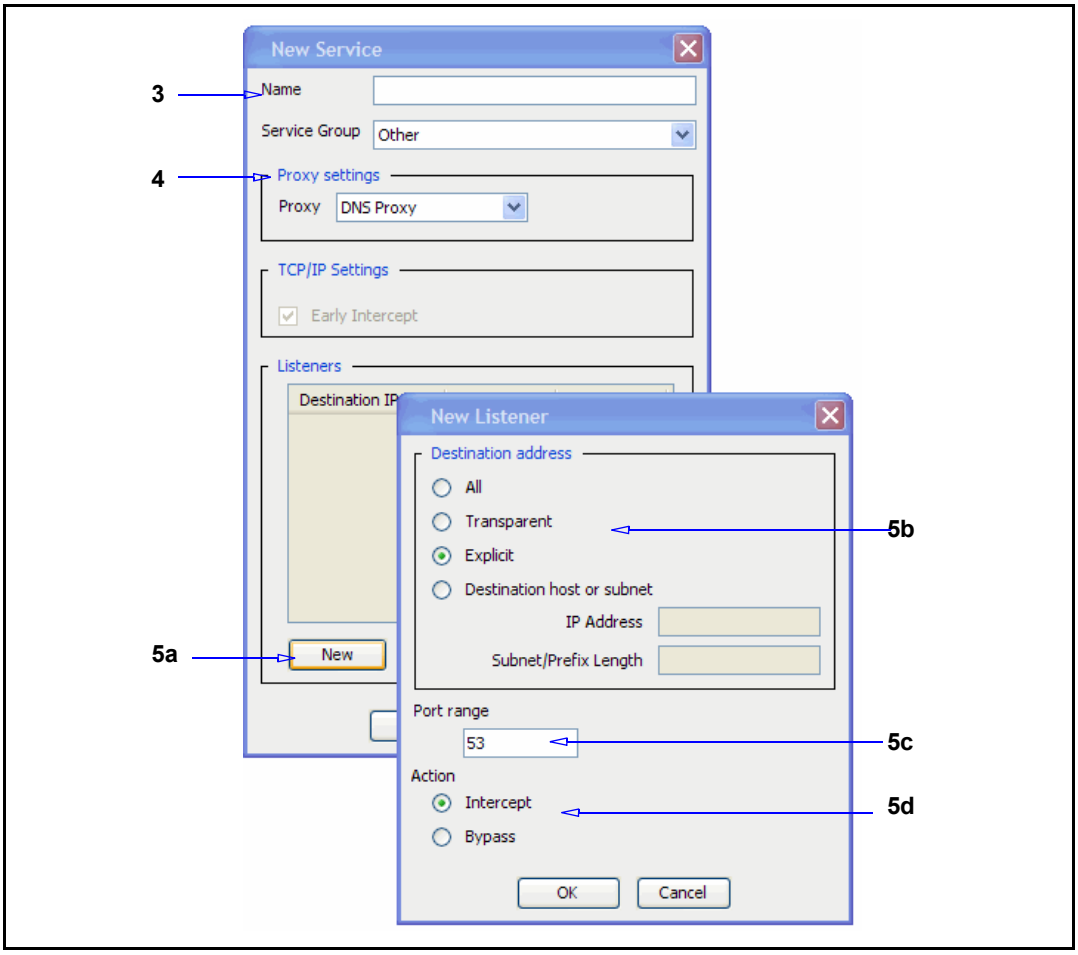


3. Scroll the list of services to display the default DNS service line; click the **+** symbol to expand the DNS services list.
4. Notice the **Action** for each default service (port 53) is **Bypass**. Select **Intercept** from the drop-down list(s).
5. Click **Apply.**

Creating or Editing a DNS Proxy Service

To create or edit a DNS proxy service:

1. From the Management Console, select **Configuration > Services > Proxy Services.**
2. To edit an existing DNS proxy service, highlight the service and click **Edit.** To create a new proxy service, click **New.**



3. In the **Name** field, choose a meaningful name for the new proxy service.
4. From the **Proxy** drop-down list, select **DNS Proxy.**
5. Create a new listener:
   a. Click **New.**
   b. Define the Destination IP information.
   c. In the **Port Range** field, enter the ports on which the service should listen. The default port is 53.
   d. Select the default action for the service: **Bypass** tells the service to ignore any traffic. **Intercept** configures the service to intercept the traffic that is being proxied.
   e. Click **OK** to close the dialog.
6. Click **Apply.**

Relevant CLI Syntax to Create/Edit a DNS Proxy Service

- To enter configuration mode for the service:

```
SGOS#(config) proxy-services
SGOS#(config proxy-services) create dns service-name
SGOS#(config proxy-services) edit service-name
```

- The following subcommands are available:

```
SGOS#(config service-name) add {transparent | explicit | all |
ip_address | ip_address/subnet-mask} {port | first_port-last_port}
[intercept | bypass]
SGOS#(config service-name) bypass {transparent | explicit | all |
ip_address | ip_address/subnet-mask} {port | first_port-last_port}
SGOS#(config service-name) exit
SGOS#(config service-name) intercept {transparent | explicit | all |
ip_address | ip_address/subnet-mask} {port | first_port-last_port}
SGOS#(config service-name) remove {transparent | explicit | all |
ip_address | ip_address/subnet-mask} {port | first_port-last_port}
SGOS#(config service-name) view
```

Creating a Resolving Name List

You can create the resolving name list that the DNS proxy uses to resolve domain names. This procedure can only be done through policy. (For a discussion on using the `<DNS-Proxy>` layer, refer to *Volume 10: Content Policy Language Guide.*)

Each name resolving list entry contains a domain-name matching pattern. The matching rules are:

- *test.com* matches only *test.com* and nothing else.
- .test.com matches *test.com, www.test.com* and so on.
- "." matches all domain names.

An optional IP address can be added, which allows the DNS proxy to return any IP address if the DNS request's name matches the domain name suffix string (*domain.name*).

To create a resolving name list, create a policy, using the <DNS-Proxy> layer, that contains text similar to the following:

```
<DNS-Proxy>
   dns.request.name=www.example.com dns.respond.a(vip)
-or-
<DNS-Proxy>
   dns.request.name=.example.com dns.respond.a(vip)
-or-
<DNS-Proxy>
   dns.request.name=www.example.com dns.respond.a(10.1.2.3)
```

Note: You can also create a resolving name list using VPM. For more information on using the DNS-Proxy layer in VPM, refer to *Volume 1: Getting Started.*

Chapter 6: Accelerating the Microsoft Outlook Application (Endpoint Mapper and MAPI Proxies)

This chapter discusses the Endpoint Mapper service and MAPI proxy, which function together to intercept traffic generated by Microsoft Outlook clients and accelerate traffic over the WAN.

Topics in this Chapter

This chapter includes information about the following topics:

- "The Endpoint Mapper Proxy Service" on page 84.
- "The MAPI Proxy" on page 93.

Section A: The Endpoint Mapper Proxy Service

This section discusses the Microsoft Remote Procedure Call (RPC) protocol and describes how to configure the Endpoint Mapper proxy service on the ProxySG.

About RPC

The Microsoft RPC protocol functions across a client/server model where one application requests a service from another application. The requesting program is the client; the providing service is the server. RPC allows an application on one host (the client) to request and thereby cause an application on another host (the server) to execute an action without the requirement of explicit code. For example: MAPI traffic.

Typically, RPC communications occur when the client contacts the Endpoint Mapper service on that client host to determine how to contact the server. The client provides the RPC service identifier and the Endpoint Mapper service returns the IP and port the client uses to contact the server. Then, the client makes a new TCP connection to that IP and port and sends its RPC request.

The challenges occur when these communications occur between branch offices and servers located in core locations. The user experience is poor because of low available bandwidth or high latency lines.

About the Blue Coat Endpoint Mapper Proxy Solution

The Endpoint Mapper proxy intercepts an RPC client request for a particular RPC service. The Endpoint Mapper proxy looks up the request in its local database, and if there is a match it replies to the client the port number the RPC service is listening on. If it is not in the database, it forwards the request up to the server. The server responds with the port number the service is listening on, and the Endpoint Mapper proxy populates its internal database. It then creates a secondary listener on that RPC port and server IP address, and responds to the RPC client with the port number. When the RPC client connects to the service, the Endpoint Mapper proxy secondary service intercepts the request and tunnels it.

Substantial performance increase occurs because:

- The ProxySG caches server information, negating the requirement to connect to an upstream server for repeated requests.
- The ProxySG at the branch compresses RPC traffic and sends it over the TCP connection to the core ProxySG, which decompresses the data before sending it to the RPC server.

The Endpoint Mapper proxy can be deployed in both transparent and explicit modes. Intercepting RPC traffic is part of the complete solution that includes the MAPI proxy ("Section B: The MAPI Proxy" on page 93).

Note: Only Microsoft RPC version 5.0 is supported. Unsupported Microsoft RPC version traffic is passed through the ProxySG without processing.

Bypassing Endpoint Mapper Traffic

Certain scenarios might require you to change the Endpoint Mapper service from **Intercept** to **Bypass**. For example, you need to take an Endpoint Mapper service offline for maintenance. When an Endpoint Mapper changes from Intercept to Bypass, the ProxySG closes not only the primary connections (such as connections to a Microsoft Exchange server on port 135), but also the secondary connections, which are used to intercept further RPC requests on mapped ports. The result is fully bypassed Endpoint Mapper traffic.

Policy Support

The Endpoint Mapper proxy supports any policy that applies to TCP tunnel connections. See "Reference: CPL Triggers, Properties, and Actions" on page 91 for supported CPL triggers and actions.

Access Logging

Each TCP connection results in an access log entry. Both the Endpoint Mapper proxy and secondary tunnel traffic activities are logged. The ProxySG main access log format is used by default.

Note: If the access log for the primary connection changes to a new log, the secondary connections are also moved to the new log.

For a reference list and descriptions of used log fields, see "Reference: Access Log Fields" on page 90.

Configuring Endpoint Mapper Service Options

This section describes how to change the default service options and add new services.

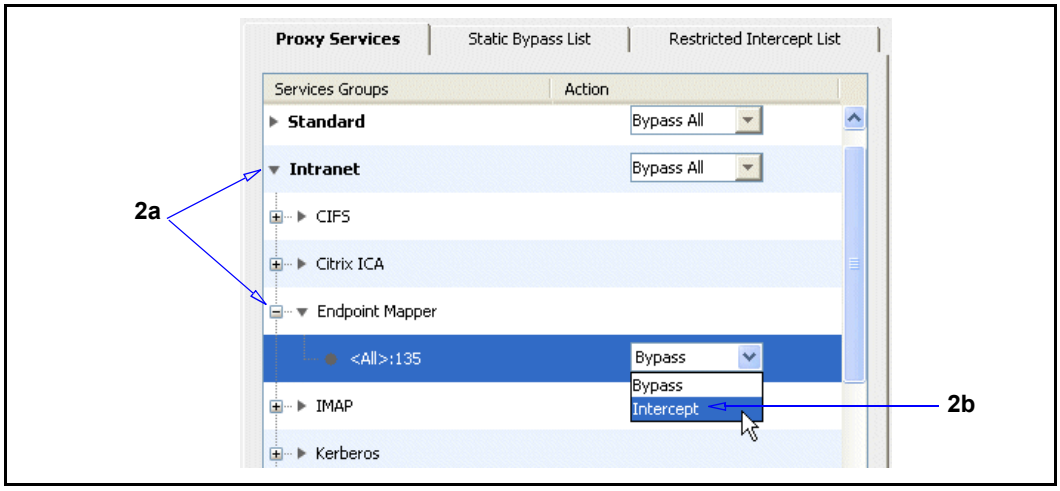
Configuring the Endpoint Mapper Service to Intercept All IP Addresses on Port 135

By default (upon upgrade and on new systems), the ProxySG has an Endpoint Mapper service configured on port `135`. The service is configured to listen to all IP addresses, but is set in **Bypass** mode.

The following procedure describes how to change the service to **Intercept** mode.

To configure the Endpoint Mapper service attributes:

1. From the Management Console, select **Configuration > Services > Proxy Services**.



2. Change the Endpoint Mapper service to Intercept:
   a. Scroll the list of services and select the **Intranet** service group; select the **Endpoint Mapper** group (the service tree expands).
   b. From the **<All>:135** drop-down list, select **Intercept.**
3. Click **Apply.**

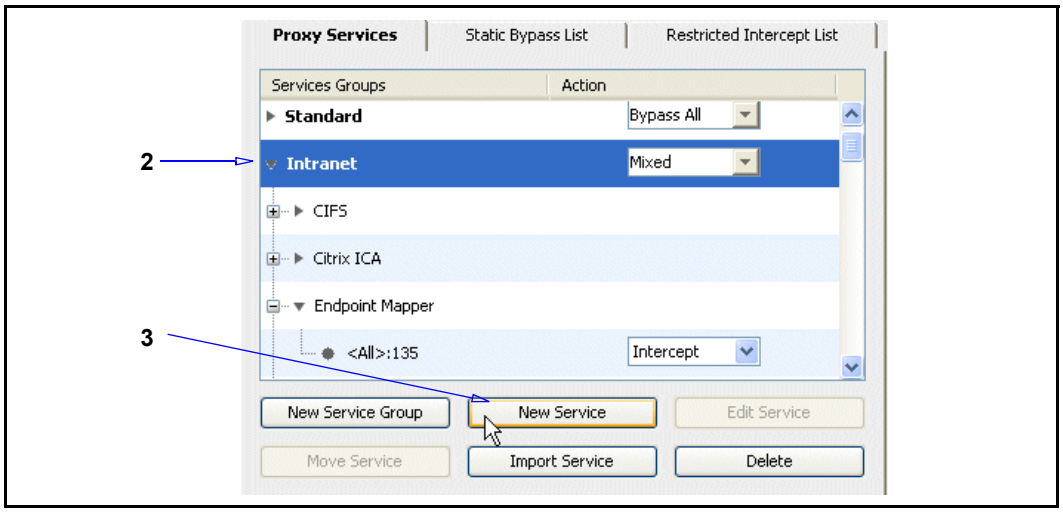
Adding a New Endpoint Mapper Service

The ProxySG allows you to add new Endpoint Mapper services. Consider the following scenario: you want the ProxySG to exclude (bypass) an IP address/subnet from MAPI acceleration because that network segment is undergoing routine maintenance.

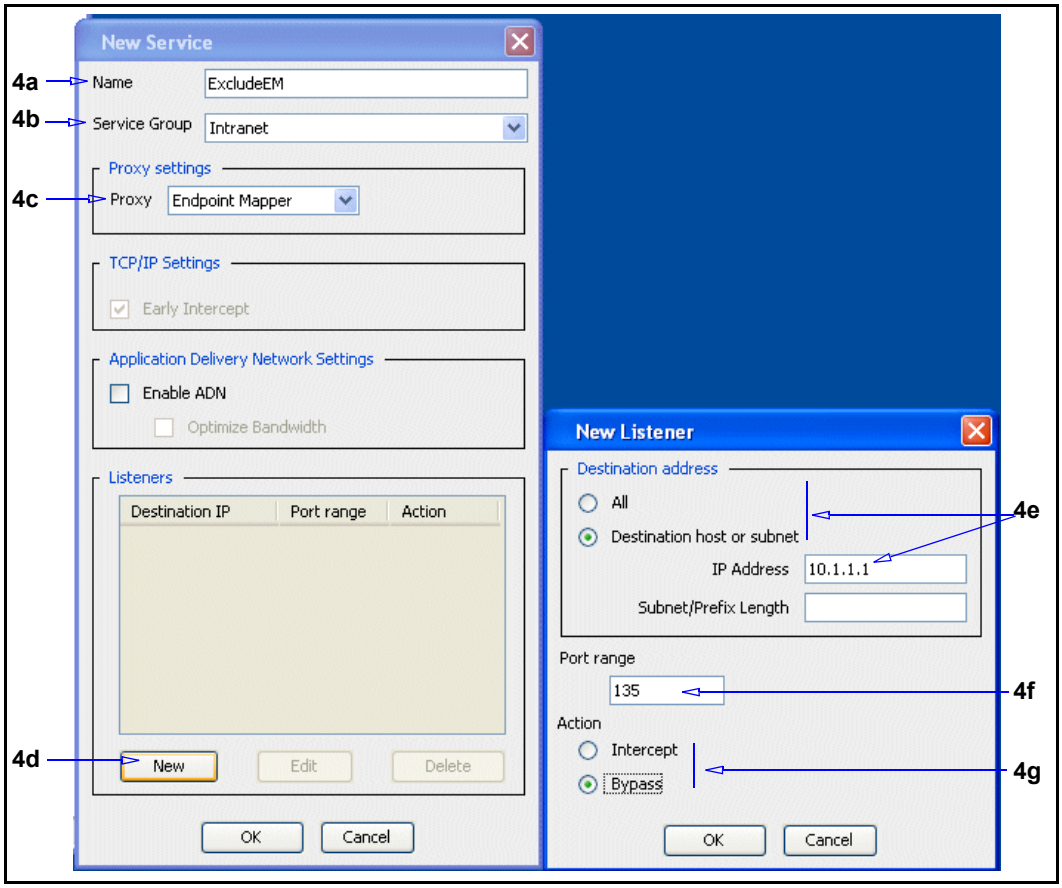
Adding a new Endpoint Mapper Service

1. From the Management Console, select **Configuration > Services > Proxy Services.**

Section A: The Endpoint Mapper Proxy Service



2. Scroll the list of services and select the **Intranet** service group; select the **Endpoint Mapper** group (the service tree expands).
3. Click **New Service**. The New Service dialog displays with some default settings.



4. Configure the service options:

   a. Name the service. In this example, the service is named **ExcludeEM** because the network admin wants to prevent the ProxySG from intercepting Endpoint Mapper traffic.

   b. From the **Service Group** drop-down list, select **Intranet**—the service group to which Endpoint Mapper traffic belongs.

   c. From the **Proxy Settings** drop-down list, select **Endpoint Mapper.**

   d. Click **New.** The New Listener dialog displays.

   e. This example selects the **Destination host or subnet** option and enters an sample IP address.

   f. This example accepts the default port of **135**. If the ProxySG is intercepting Endpoint Mapper traffic on a different port, the port must specified here.

   g. This example selects **Bypass** as the option; the ProxySG does not intercept Endpoint Mapper traffic.

   h. Click **OK** in each dialog to close them. The new service displays under the **Intranet** service group as its own service, not under the **Endpoint Mapper** service.

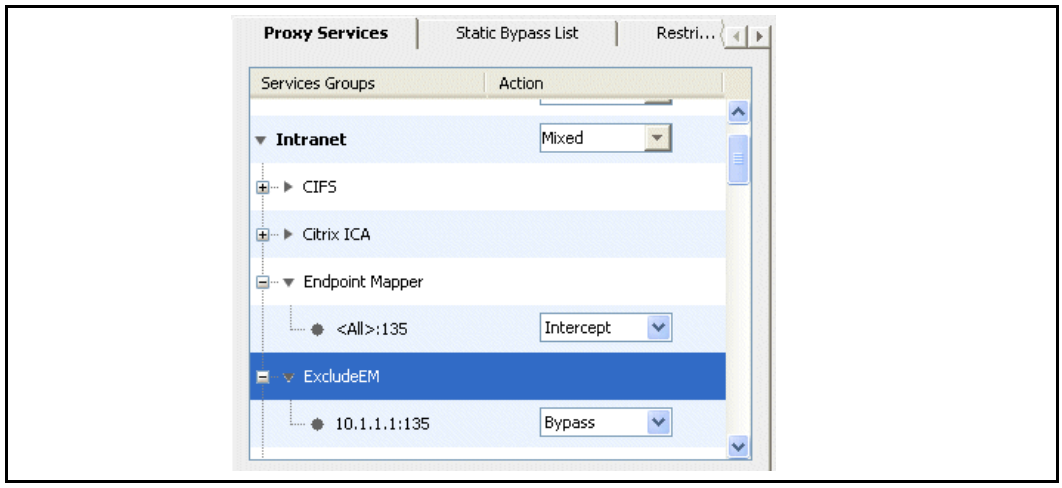


Figure 6–1 The new service displays.

Verifying the New Service

The next section references Endpoint Mapper statistics. For this example, refer to the Active Sessions statistics, from which you can view bypassed connections.

Reviewing Endpoint Mapper Proxy Statistics

After RPC traffic begins to flow through the ProxySG, you can review the statistics page and monitor results in various categories. The presented statistics are representative of the client perspective.

Management Console Statistics Pages

Endpoint Mapper statistics display across multiple pages:

- **Statistics > Traffic Mix** tab—Service and proxy data; bandwidth use and gain; client, server, and bypassed bytes. Includes all traffic types, but you can limit the scope to Endpoint Mapper data.
- **Statistics > Traffic History** tab—Service and proxy data; bandwidth use and gain; client, server, and bypassed bytes. Select Endpoint Mapper service or proxy (related to MAPI, as described in Section B: "The MAPI Proxy" on page 93).
- **Statistics > Active Sessions**—The **Proxied Sessions** and **Bypassed Connections** tabs display statistics filtered by various criteria, such as port or service type (select **Endpoint Mapper**).

Statistic URL Pages

Endpoint Mapper proxy statistics pages are viewable from Management Console URLs.

Statistics

This page displays various, more granular connection and byte statistics.

`https://SG_IP_address:8082/epmapper/statistics`

Detailed Statistics

This page displays specific client-to-server connection and file information and statistics.

```
https://SG_IP_address:8082/epmapper/detailed-statistics
```

Reference: Equivalent Endpoint Mapper Proxy CLI Commands

The Management Console procedures in this section have the following equivalent CLI command roots:

- To enter configuration mode for the service:

```
SGOS#(config) proxy-services
SGOS#(config proxy-services) create endpoint-mapper service-name
SGOS#(config proxy-services) edit service-name
```

- The following subcommands are available:

```
SGOS#(config service-name) add {all | ip_address | ip_address/subnet-mask} {port | first_port-last_port} [intercept | bypass]
SGOS#(config service-name) attribute {adn-optimize {enable | disable}| use-adn {enable | disable}}
SGOS#(config service-name) bypass {all | ip_address | ip_address/subnet-mask} {port | first_port-last_port}}
SGOS#(config service-name) exit
SGOS#(config service-name) intercept {all | ip_address | ip_address/subnet-mask} {port | first_port-last_port}}
SGOS#(config service-name) remove {all | ip_address | ip_address/subnet-mask} {port | first_port-last_port}
SGOS#(config service-name) view
```

Reference: Access Log Fields

The default ProxySG Endpoint Mapper Access Log fields are:

- `date`: GMT Date in `YYYY-MM-DD` format.
- `time`: GMT time in `HH:MM:SS` format.
- `cs-bytes, sr-bytes, rs-bytes, sc-bytes:` Standard ELFF format. The total RPC byte counts in the specified direction (client-server).
- `cs-method`: Request method used from client to appliance.
- `time-taken`: Time taken (in milliseconds) to process the request.
- `c-ip`: IP address of the RPC client.
- `s-action`: The logging action (or flow) being one of the following:
  - `ALLOWED`: Endpoint operation passed the policy checkpoint and was also successful.
  - `DENIED`: Endpoint operation failed the policy checkpoint.

  - FAILED: Endpoint operation was successful on the server but failed on the proxy for some internal reason.
  - TUNNELED: Traffic was tunneled.
- cs-uri-scheme: Scheme from the log URL.
- cs-host: Hostname from the client's request URL. If URL rewrite policies are used, this field's value is derived from the log URL.
- cs-port: Port used from the client to the appliance.
- cs-username: Relative username of a client authenticated to the proxy (for example: not fully distinguished).
- s-supplier-ip: IP address of the upstream host (not available for a cache hit).
- s-supplier-name: Hostname of the upstream host (not available for a cache hit).
- s-supplier port: Port number of the upstream host (not available for a cache hit).
- r-supplier-ip: IP address used to contact the upstream host (not available for a cache hit).
- r-supplier-name: Hostname used to contact the upstream host (not available for a cache hit).
- r-supplier port: Port used to contact the upstream host (not available for a cache hit).
- sc-filter-result: Content filtering result: Denied, Proxied, or Observed.
- sc-filter-category: Content filtering category.
- s-ip: IP address of the appliance on which the client established its connection.
- s-sitename: Service used to process the transaction.

Reference: CPL Triggers, Properties, and Actions

The following ProxySG CPL is supported in the Endpoint Mapper proxy service:

- allow / deny

TCP Tunneling Triggers

- Client: client.address, client.host, client.host.has_name, client protocol (recognizes epmapper token).
- Date / Time: date[.utc], day, hour, minute, month, weekday, year, time
- Proxy: proxy.address, proxy.port, proxy.card
- has_client
- url

Section A: The Endpoint Mapper Proxy Service

Properties and Actions

- allow/deny
- trace
- log_message
- notify_email, notify_snmp
- reflect_ip
- access_log
- forward
- socks_gateway

Section B: The MAPI Proxy

This section discusses the Messaging Application Programing Interface (MAPI) protocol and describes how to configure the services and proxy on the ProxySG.

About MAPI

MAPI is the protocol used by Microsoft Outlook (client) to communicate with Microsoft Exchange (server), most commonly for e-mail applications. MAPI itself is based on the Microsoft Remote Procedure Call (RPC).

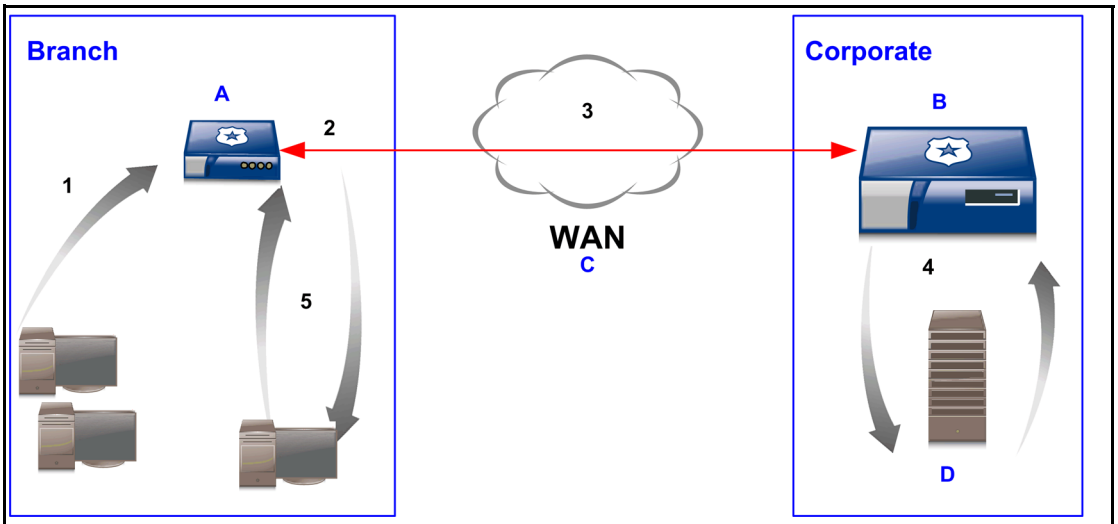
Because MAPI is based on RPC, it suffers from the same performance inherent with RPC communications. Microsoft Outlook is the most common enterprise e-mail application. As enterprises continue to trend toward consolidating servers, which requires more WAN deployments (branch and remote locations), e-mail application users experience debilitating response times for not only sending and receiving mail, but accessing message folders or changing calendar elements. This is because MAPI RPC transmissions are broken into *blocks* of data (no more than 32 KB). The client must stop and wait for each block to arrive before requesting the next block. Each stop represents time lost instead of data sent.

About the Blue Coat MAPI Solution

The MAPI proxy is similar to and actually works in conjunction with the Endpoint Mapper proxy in that it intercepts and accelerates RPCs; however, MAPI is always deployed transparently and does *not* listen on a specific port or port range. Instead, when configured to do so, the Endpoint Mapper proxy *hands off* Outlook / Exchange traffic to the MAPI proxy (but the Endpoint Mapper proxy functionality is still required to make an RPC connection).

The MAPI proxy itself is a *split proxy*, which is only viable in a deployment that consists of a branch proxy and core proxy. A split proxy employs co-operative processing at the branch and the core to implement functionality that is not possible in a standalone proxy. In the case of the MAPI Proxy, cooperation exists between the branch ProxySG and the core ProxySG to reduce the number of RPCs sent across the WAN.

The TCP connection between the branch and core proxies makes use of byte caching for acceleration (compression).



LEGEND:

A: A ProxySG 210 at a branch office; Endpoint Mapper proxy is configured on 135; MAPI proxy: MAPI handoff, batching, and keep-alive are enabled.

B: A ProxySG 8100 appliance (concentrator) at a corporate location.

C: Wide Area Network (Internet); the ProxySG appliances communicate through a TCP tunnel.

D: Microsoft Exchange server.

PROCESS FLOW:

1: During business hours, two branch Microsoft Outlook clients send e-mails with attachments.

2: The branch proxy batches RPC messages into larger chunks.

3: With the default Endpoint Mapper proxy configuration, byte caching compresses the data over the TCP connection.

4: The core proxy performs decompression and connects to the Exchange server for processing to destination client.

5. Another user logs out of Microsoft Outlook at the end of the day. With keep-alive configured, the ProxySG maintains a connection to the Exchange server and continues to queue sent mail. When the user logs in the next morning, the ProxySG delivers the cached mail, eliminating excessive WAN traffic increase.

Figure 6–2 MAPI Proxy Deployment and Flow Diagram

Reducing RPC Messages Across the WAN

The MAPI proxy Batching feature reduces the number of RPC messages traversing the WAN. The branch and core appliances work together to batch multiple RPC messages in a larger chunk, rather than sending the smaller chunks. Also, the proxy *predicts*, or reads ahead, what will be requested next. When the branch proxy receives data chunks, it begins sending the acknowledgments to the MAPI client to satisfy that requirement of the communication process.

Maintaining Exchange Connections

The MAPI proxy Keep-Alive feature allows the ProxySG to maintain the connection to the Exchange server *after* the user has logged off from Outlook. Determined by the configurable interval, the MAPI proxy checks the Exchange server for new mail. *ADN Optimization* allows the connection to remain *warm* so that when the user logs on again to Outlook, the number of retrieved bytes is lower, allowing for better performance.

The MAPI proxy remembers *each* user that is logged on or off. If the duration exceeds the specified limit, or when the user logs back into the mail application, the Keep-Alive connection is dropped.

For more information about ADN optimization, refer to *Volume 5: Advanced Networking*.

Supported Servers

The MAPI proxy supports protocol optimization, byte caching, and compression for:

- MAPI 2000—Any Outlook client to Exchange Server 2000.
- MAPI 2003—Outlook client 2003 and above to Exchange Server 2003.

The MAPI proxy supports byte caching, and compression for:

- MAPI 2007—Outlook client 2007 to Exchange Server 2007.

Access Logging

The MAPI proxy uses a default access log format. Data includes user actions, data lengths (bytes), and RPC data.

```
date, time, c-ip, c-port, r-ip, r-port, x-mapi-user, x-mapi-method,
cs-bytes, sr-bytes, rs-bytes, sc-bytes, x-mapi-sc-rpc-count, x-mapi-
sr-rpc-count, x-mapi-rs-rpc-count, x-mapi-sc-rpc-count, s-action, cs-
username, cs-auth-group, s-ip
```

For MAPI-specific descriptions, see "Reference: Access Log Fields" .

More Conceptual Reference

- "About RPC" on page 84.
- *Volume 5: Advanced Networking.*

Configuring the ProxySG MAPI Proxy

This section contains the following sub-sections:

- "This section describes how to change the default service options and add new services." on page 85.
- "Reviewing Endpoint Mapper Proxy Statistics" on page 89.

About the MAPI Service

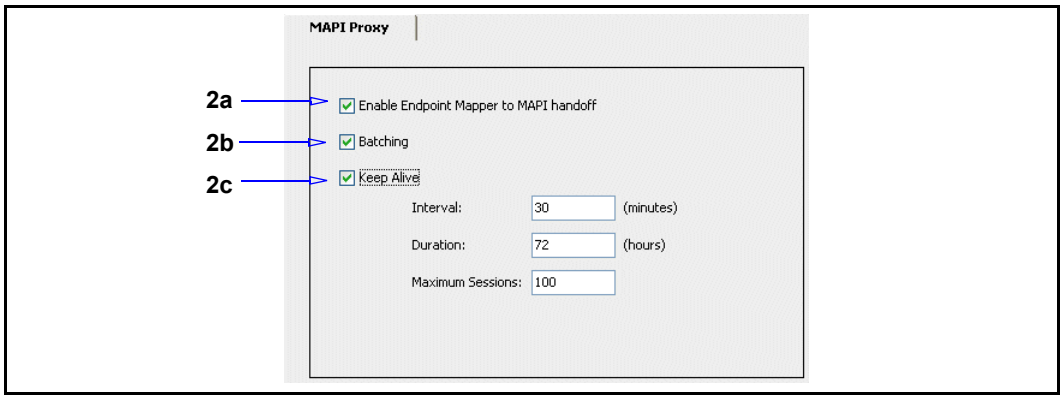
By default (upon upgrade and on new systems), the ProxySG has an Endpoint Mapper proxy service configured on port 135. The service is also configured to listen to all IP addresses, but is set in **Bypass** mode. As the MAPI proxy processes RPC communication as well, it uses the Endpoint Mapper proxy service. See "Section A: The Endpoint Mapper Proxy Service" on page 84.

Configuring the MAPI Proxy

The MAPI Proxy options concern Batching, Handoff, and Keep-Alive features. This section describes these options and why they might require changing based on your branch deployment.

To view/change the MAPI Proxy configuration options:

1. In the Management Console, select **Configuration > Proxy Settings > MAPI Proxy.**



2. Configure the MAPI Proxy configuration options:

   a. **Enable Endpoint Mapper to MAPI Handoff**: The Endpoint Mapper proxy sends Microsoft Outlook and Exchange RPC communications to the MAPI proxy, which is used to manage the data. The routing connections from the branch to the core remains under the control of the Endpoint Mapper service.

   Note: A secondary TCP connection is created to handle all non-MAPI traffic. No changes to the Endpoint Mapper service or proxy are required.

   b. **Batching**: If enabled, MAPI traffic across the WAN is accelerated because data is chunked and sent as one connection, rather than multiple smaller connections.

   c. **Keep-Alive**: After a user logs out of Outlook, the MAPI RPC connection remains and the ProxySG continues to receive incoming messages to this account. If disabled (the default), no attempts to contact the server occur until the next time the user logs into his/her Outlook account. This might create a noticeable decrease in performance, as the queue of unreceived mail is processed.

- **Interval**: If **Keep-Alive** is enabled, how often the MAPI proxy contacts the Exchange server to check for new messages.
- **Duration**: If **Keep-Alive** is enabled, how long the MAPI proxy maintains the connection to the Exchange server. The connection is dropped if the duration exceeds this value or once a user logs back in to the mail application.
- **Maximum Sessions**: Limits the number of occurring active keep-alive sessions. If a new keep-alive session starts, and the specified limit is already exceeded, the oldest keep-alive session is *not* dropped but no new keep-alive sessions are created.

3. Click **OK**; click **Apply**

Reviewing MAPI Statistics

After MAPI traffic begins to flow through the ProxySG, you can review the statistics page and monitor results in various MAPI categories. The presented statistics are representative of the client perspective.

To review MAPI History:

1. From the Management Console, select **Statistics > MAPI History.**
2. View statistics:
   a. Select a statistic category tab:
      - **MAPI Clients Bytes Read**: The total number of bytes read by MAPI clients.
      - **MAPI Clients Bytes Written**: The total number of bytes written by MAPI clients.
      - **MAPI Clients**: The total number of connected MAPI clients.
   b. The graphs display three time metrics: the previous 60 minutes, the previous 24 hours, and the previous month. Roll the mouse over any colored bar to view the exact metric.
3. (Optional) You can change the scale of the graph to display the percentage of bar peaks to display.

To review MAPI Active Sessions:

1. From the Management Console, select the **Statistics > Active Sessions > Proxied Sessions** tab.
2. From the first **Filter** drop-down list, select **Proxy**; from the second drop-down list, select **MAPI**.
3. Click **Show.** The **Proxied Sessions** area displays MAPI statistics.

Reference: Equivalent MAPI Proxy CLI Commands

The Management Console procedures in this chapter have the following equivalent CLI command roots:

```
SGOS#(config) mapi
```

- The following subcommands are available:

```
SGOS#(config mapi) handoff (enable | disable}
SGOS#(config mapi) batching {enable | disable}
SGOS#(config mapi) keep-alive {enable | disable}
SGOS#(config mapi) keep-alive interval [minutes 1-60]
SGOS#(config mapi) keep-alive duration [hours 1-72]
SGOS#(config mapi) {view | exit}
```

Reference: Access Log Fields

The default MAPI Access Log fields are:

```
"date time c-ip c-port r-ip r-port x-mapi-user "\
"x-mapi-method cs-bytes sr-bytes rs-bytes sc-bytes "\
"x-mapi-cs-rpc-count x-mapi-sr-rpc-count "\
"x-mapi-rs-rpc-count x-mapi-sc-rpc-count "\
"s-action cs-username cs-auth-group s-ip"
```

- `cs-bytes, sr-bytes, rs-bytes, sc-bytes`: Standard ELFF format. The total RPC byte counts in the specified direction (client-server).
- `x-mapi-method`: The end-user operation, one of:
  - `STARTUP`: The start of a MAPI session. A single user can have more than one active MAPI sessions for a single instance of Outlook.
  - `SHUTDOWN`: The end of a MAPI session.
  - `SEND`: Outlook is sending an e-mail (with or without attachments) to Exchange and the ProxySG is batching the contents.
  - `FETCH`: Outlook is fetching an e-mail (with or without attachments) to Exchange and the ProxySG is batching the contents.
  - `KEEP_ALIVE_STARTUP`: A keep-alive session started.
  - `KEEP_ALIVE_SHUTDOWN`: A keep-alive session ended.
  - `KEEP_ALIVE_NEGOTIATE`: Messages were sent to query the state of the Inbox during a keep-alive session.
  - `KEEP_ALIVE_FETCH`: An e-mail (with or without attachments) was fetched during a keep-alive session.
- `x-mapi-user-dn`: The full user domain name gathered from the MAPI negotiation of user credentials between Outlook and Exchange.
- `x-mapi-user`: A shortened form of the user domain name.
- `s-action`:
  - `ALLOWED`: The traffic was permitted through.
  - `SUCCESS`: The traffic was successfully proxied, but was not subject to policy.
  - `DENIED`: The traffic was denied by policy.
  - `SERVER_ERROR`: The traffic was dropped because of an error communicating with the server.

- `FAILED`: The traffic was dropped because of an error when handling the messages sent by the client. Or an internal problem with the MAPI proxy.
- `BATCHED`: The traffic was batched.
- `TUNNELED`: The traffic was tunneled to the Exchange server for one of two reasons:
  - The MAPI traffic is encrypted; therefore, the ProxySG cannot batch messages or attachments and thus cannot provide WAN optimization benefits.
  - The MAPI proxy could not connect upstream through an Application Delivery Network (ADN) tunnel.

- `x-cs-mapi-rpc-count`: The number of RPCs sent from the client to the proxy.
- `x-sr-mapi-rpc-count`: The number of RPCs sent from the proxy to the server.
- `x-rs-mapi-rpc-count`: The number of RPCs sent from the server to the proxy.
- `x-sc-mapi-rpc-count`: The number of RPCs sent from the proxy to the client.

Chapter 7: Managing the File Transport Protocol (FTP) Proxy

This chapter discusses the Blue Coat implementation of proxy support for File Transport Protocol (FTP).

The ProxySG supports two FTP modes:

- Native FTP, where the client connects through the FTP proxy. Native FTP is used when the client connects (either explicitly or transparently) through FTP; the ProxySG then connects upstream through FTP (if necessary).
- Web FTP, where the client uses an explicit HTTP connection. Web FTP is used when a client connects in explicit mode using HTTP and accesses an `ftp://` URL. The ProxySG translates the HTTP request into an FTP request for the origin content server (OCS), if the content is not already cached, and then translates the FTP response with the file contents into an HTTP response for the client.

In most environments, such as one with an inline deployment, the FTP proxy defaults are satisfactory and do not need to be changed. If, however, your environment has the proxy transparently intercepting only FTP control traffic, you might need to modify the method the ProxySG uses to determine IP addresses for FTP data connections.

Topics in this Chapter

This chapter includes information about the following topics:

- "How Do I...?"
- "About FTP" on page 102
- "Configuring the ProxySG for Native FTP Proxy" on page 104
- "Configuring FTP Connection Welcome Banners" on page 109
- "Viewing FTP Statistics" on page 110

How Do I...?

To use this chapter, identify the task and click the link:

How do I...?	See...
Understand how the ProxySG manages IP addresses?	"About FTP" on page 102
Configure IP addresses?	"Configuring IP Addresses for FTP Control and Data Connections" on page 102
Configure native FTP?	"Configuring the ProxySG for Native FTP Proxy" on page 104

How do I...?	See...
Configure Web FTP?	Chapter 8: "Intercepting and Optimizing HTTP Traffic" on page 111
Customize the *welcome banner* for the FTP proxy?	"Configuring FTP Connection Welcome Banners" on page 109
View FTP statistics?	"Viewing FTP Statistics" on page 110

About FTP

With Blue Coat's implementation of FTP, you can control how the ProxySG responds to FTP client requests. You can also control which IP addresses are used for control and data connections to the server.

This section discusses:

- "Terminology"
- "Configuring IP Addresses for FTP Control and Data Connections" on page 102
- "Client-Side Data Connections Mode" on page 103
- "FTP Server Notes" on page 104

Terminology

- FTP control and data connections: FTP uses two parallel TCP connections to transfer a file, a control connection and a data connection.
  - control connections: Used for sending control information, such as user identification and password, between two hosts.
  - data connections: Used to send the file contents between two hosts. By default, the ProxySG allows both active and passive data connections.
    - Active (PORT) mode data connections: Data connections initiated by an FTP server to an FTP client at the port and IP address requested by the FTP client. This type of connection method is useful when the FTP server can connect directly to the FTP client.
    - Passive (PASV) mode data connections: Data connections initiated by an FTP client to an FTP server at the port and IP address requested by the FTP server. This type of connection is useful in situations where an FTP server is unable to make a direct connection to an FTP client because the client is located behind a firewall or other similar device where outbound connections from the client are allowed, but inbound connections to the client are blocked.

Configuring IP Addresses for FTP Control and Data Connections

The FTP client determines whether the client-side data connection is active or passive from the client to the ProxySG. The ProxySG determines the server-side connections.

By default, the ProxySG allows both active and passive data mode connections. FTP connections are divided into client-side control and data connections and server-side control and data connections.

- Client-side control connection: The proxy always uses the IP address selected by the client to respond to the client.
- Client-side data connection: The proxy's behavior depends on the `ftp.match_client_data_ip(yes | no)` property. If this property is enabled (the default), the proxy uses the same IP address for the data connection as it used for the client-side control connection. If the property is disabled, the proxy uses its own IP address, choosing the address associated with the interface used to connect back to the client.
- server-side control connection: The proxy uses the IP address selected by the `reflect_ip(auto | no | client | vip | ip_address)` property. By default, this is the local proxy IP address associated with the interface used to connect to the server.

  Client IP reflection is set globally from the **Configuration > Proxy Settings > General** tab. By default, the CPL `reflect_ip( )` setting is `auto`, which uses this global configuration value.

  Note: Setting client IP address reflection for FTP affects the source address that is used when making the outgoing control connection to the origin server. It might also affect which address is used by the proxy for data connections.

- Server-side data connection: The proxy's behavior depends on the `ftp.match_server_data_ip(yes | no)` property. If this property is enabled (the default), the proxy uses the same IP address for the data connection as it used for the server-side control connection. If the property is disabled, the proxy uses its own IP address to communicate with the server, choosing the address associated with the interface used to connect to the server.

  Note: Either the `reflect_ip( )` property or the `reflect-client-ip` configuration must be set for the `ftp.match_server_data_ip(yes)` property to be meaningful.

For information on creating and modifying policy through VPM, refer to *Volume 6: The Visual Policy Manager and Advanced Policy*. For information on creating and modifying policy through CPL, refer to *Volume 10: Content Policy Language Guide*. The `ftp.match_server_data_ip( )` and `ftp.match_client_data_ip( )` properties can only be set through CPL.

Client-Side Data Connections Mode

Administrators determine how the ProxySG responds to a request from an FTP client for a passive mode data connection (PASV command).

By default, some FTP clients do not open a passive mode data connection to an IP address that is different from the IP address used for the control connection.

When PASV is disabled, some FTP clients try a PORT command automatically, which allows requests to be received when the client doesn't allow PASV connections to a different IP address.

Note: Some clients might display an error when PASV is disabled on the ProxySG, requiring you to manually request PORT mode.

The FTP client software controls any messages displayed to the end user as a result of this response from the ProxySG.

Server-Side Data Connections Mode

The `ftp.server_data(auto | passive | port)` property controls the type of server-side data connection that the ProxySG opens to the server. The default of `auto` means to try a passive connection first and then fall back to an active connection if that fails.

FTP Server Notes

IIS and WS_FTP servers do not support:

- Passive data connections with a source IP address that is different from the source IP address of the control connection.
- Active data connections with a destination IP address that differs from the source IP address of the control connection.

The `ftp.match_server_data_ip(no)` property most likely will not work correctly with these servers.

Notes

- Internet Explorer does not support proxy authentication for native FTP.
- The FTP proxy does not support customized exception text; that is, you can use policy to deny requests, but you can't control the text sent in the error message.

Configuring the ProxySG for Native FTP Proxy

This section discusses:

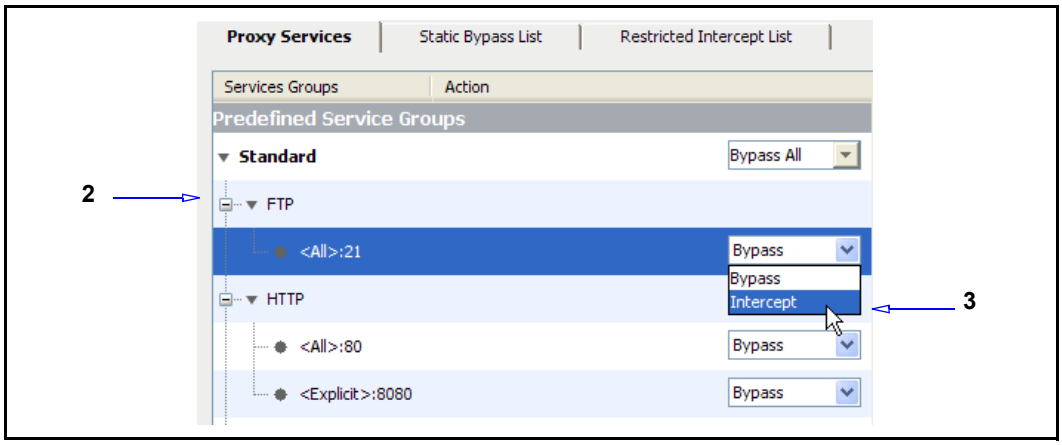
- "Creating or Editing the FTP Service"
- "Configuring the FTP Proxy" on page 107
- "Configuring FTP Clients" on page 108

Changing the Default FTP Proxy Service to Intercept

By default services are configured to accept all IP addresses in **Bypass** mode. The following procedure describes how to change them to **Intercept** mode, and explains other attributes within the service.

To configure the FTP proxy to intercept file sharing traffic:

1. From the Management Console, select **Configuration > Services > Proxy Services**.



2. Scroll the list of services to display the default FTP service line; click the **+** symbol to expand the FTP services list.
3. Notice the **Action** for each default service (port 21) is **Bypass**. Select **Intercept** from the drop-down list(s).
4. Click **Apply**.

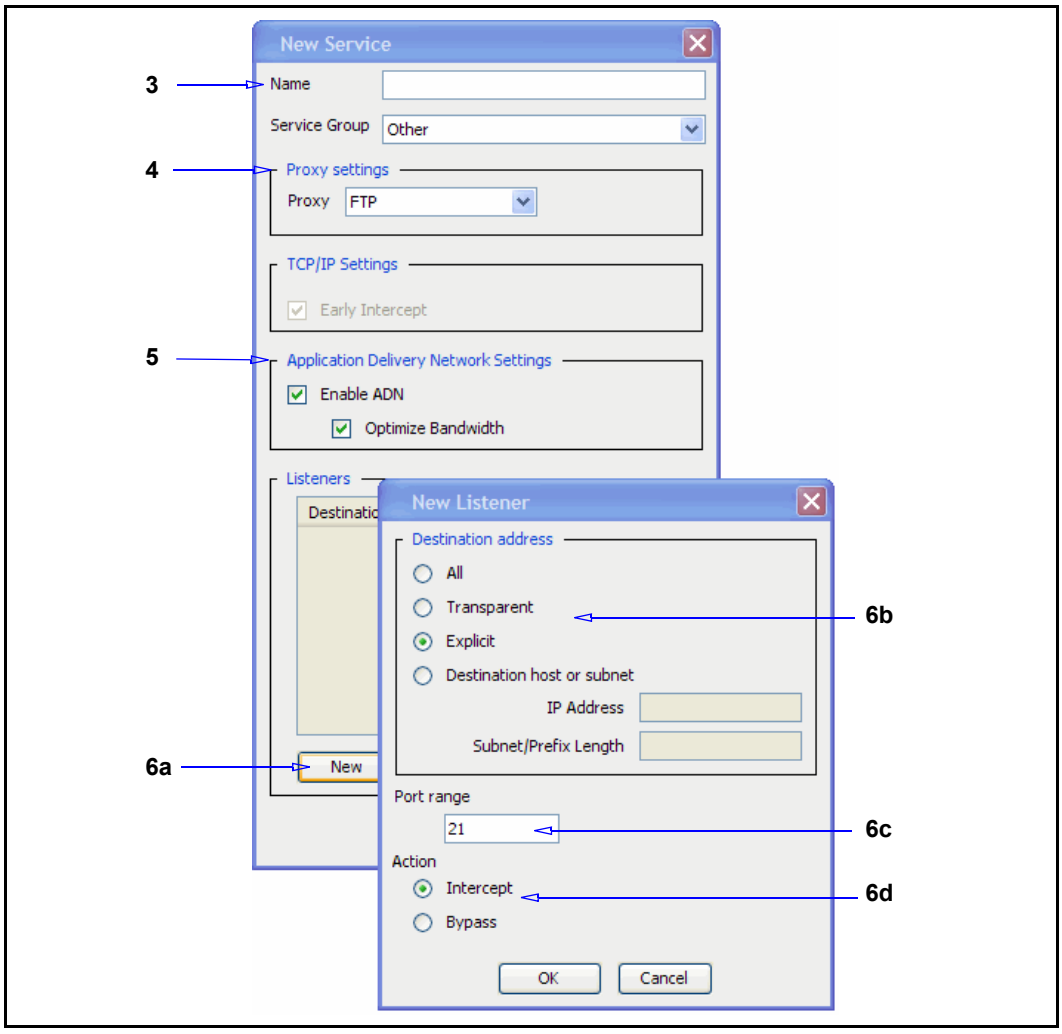
Creating or Editing the FTP Service

An FTP service is created by default, but it is in bypass mode. The service is not functioning until it is in intercept mode.

Note: Web FTP requires an HTTP service, not an FTP service. For information on configuring an HTTP proxy service, see Chapter 8: "Intercepting and Optimizing HTTP Traffic" on page 111.

To create or edit an FTP proxy service:

1. From the Management Console, select **Configuration > Services > Proxy Services**.
2. To edit an existing FTP proxy service, highlight the service and click **Edit**. To create a new proxy service, click **New**.



3. If you are creating a new FTP proxy service, enter a meaningful name in the **Name** field.
4. Select FTP from the drop-down list under **Proxy settings**.
5. Configure ADN options:
    a. The **Enable ADN** controls whether ADN is enabled for a specific service. Enabling ADN does not guarantee the connections are accelerated by ADN. The actual enable decision is determined by ADN routing (for explicit deployment) and network setup (for transparent deployment).

Note: ADN supports passive FTP (the data connection is initiated by an FTP client to an FTP server at the port and IP address requested by the FTP server. Active FTP, where data connections are initiated by an FTP server to an FTP client at the port and IP address requested by the FTP client, is not supported.

b. The **Optimize Bandwidth** checkbox is selected by default if you enabled ADN optimization during initial configuration. De-select the checkbox if you are not configuring ADN optimization.

6. Create a new listener:
   a. Click **New.**
   b. Select a Destination IP option.
   c. In the **Port Range** field, enter the ports on which the service should listen. The default port for FTP is 21.
   d. Select the default behavior for the service: **Bypass** tells the service to ignore any traffic. **Intercept** configures the service to intercept the traffic that is being proxied.
   e. Click **OK** to close the dialog.
7. Click **Apply.**

Related CLI Syntax to Create/Edit an FTP Proxy Service

- To enter configuration mode for the service:

```
SGOS#(config) proxy-services
SGOS#(config proxy-services) create ftp service-name
SGOS#(config proxy-services) edit service-name
```

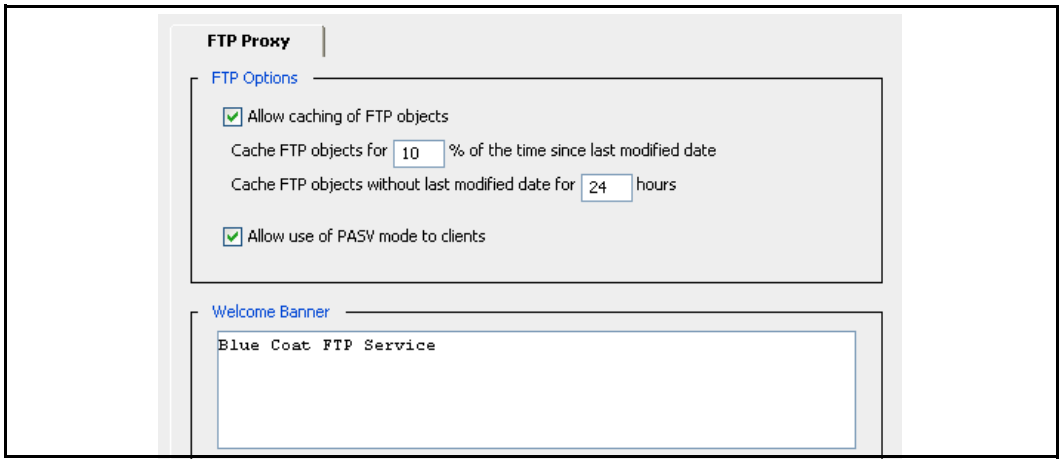
- The following subcommands are available:

```
SGOS#(config service-name) add {all | ip_address | ip_address/subnet-
mask} {port | first-port_last-port} [intercept | bypass]}
SGOS#(config service-name) attribute adn-optimize {enable | disable}|
| use-adn {enable | disable}
SGOS#(config service-name) bypass {all | ip_address | ip_address/
subnet-mask} {port | first-port_last-port}
SGOS#(config service-name) exit
SGOS#(config service-name) intercept {all | ip_address | ip_address/
subnet-mask {port | first-port_last-port}
SGOS#(config service-name) remove {all | ip_address | ip_address/
subnet-mask {port | first-port_last-port}
SGOS#(config service-name) view
```

Configuring the FTP Proxy

To configure the FTP proxy:

1. Select **Configuration > Proxy Settings > FTP Proxy.**



2. Select **Allow caching of FTP objects**. The default is enabled.
3. Determine the amount of time in percentage of how long since the object was last modified. The default is 10%.
4. Enter an amount, in hours, that the object remains in the cache before becoming eligible for deletion. The default is 24 hours.
5. Select **Allow use of PASV mode to clients**. The default is **enabled**, allowing data connections to be initiated by an FTP client to an FTP server at the port and IP address requested by the FTP server.

Related CLI Syntax to Configure the FTP Proxy

```
SGOS#(config) ftp login-syntax {raptor | checkpoint}
SGOS#(config) ftp passive-mode {enable | disable}
SGOS#(config) ftp no welcome-banner
SGOS#(config) ftp welcome banner
SGOS#(config) caching
SGOS#(config caching) max-cache-size number8
SGOS#(config caching) ftp
SGOS#(config caching ftp) enable
SGOS#(config caching ftp) type-m-percent number
SGOS#(config caching ftp) type-n-initial number
```

Note: Neither proxy authentication for transparent FTP nor proxy chaining are supported with the Checkpoint syntax.

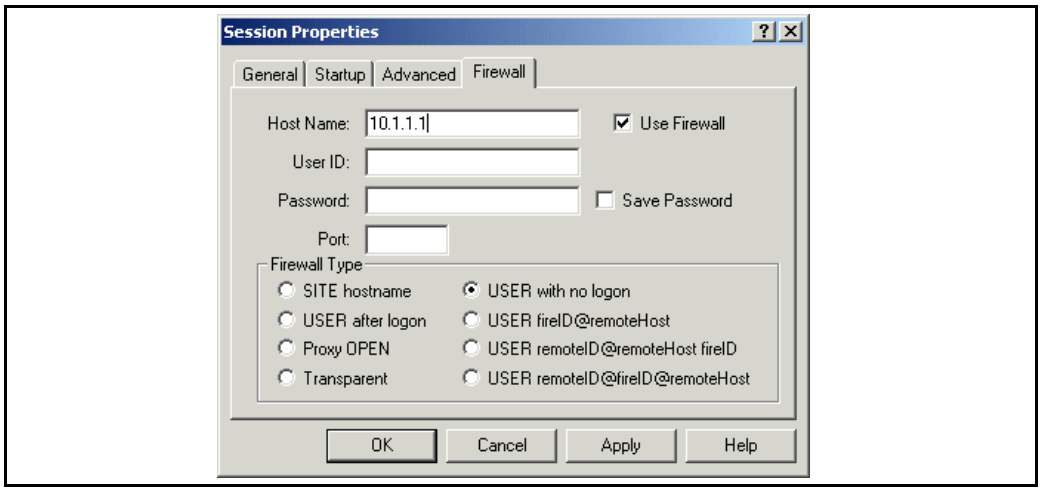
Configuring FTP Clients

Configure an FTP client to explicitly proxy to the ProxySG. Complete the following steps if you have a WSFtp client. Other clients vary.

- Enable firewall.
- Select **USER with no logon** unless you are doing proxy authentication. In that case, select **USER remoteID@remoteHost fireID** and specify a proxy username and password.

Example

The illustration demonstrates a WSFtp client configuration.



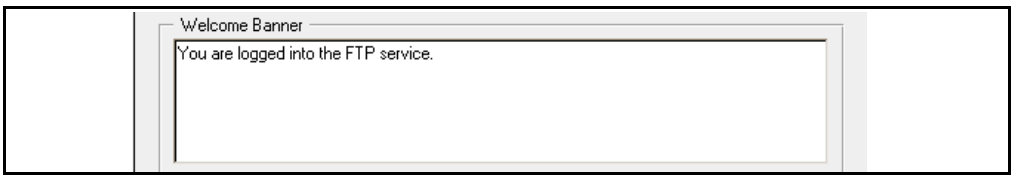
Configuring FTP Connection Welcome Banners

You can customize banners that usually describe the policies and content of the FTP server displayed to FTP clients. Without modification, the ProxySG sends a default banner to newly-connected FTP clients: **Welcome to Blue Coat FTP**. However, you might not want users to know that a ProxySG exists on the network. A default banner can be defined in the Management Console or the CLI, but other banners defined for specific groups can be created in policy layers.

Note: Configurable banners are only displayable when FTP is explicitly proxied through the ProxySG. In transparent deployments, the banner is sent to the client when proxy authentication is required; otherwise, the banner is forwarded from the FTP server.

To define the default FTP banner:

1. Select **Configuration > Services > FTP Proxy.**
2. In the **Welcome Banner** field, enter a line of text that is displayed on FTP clients upon connection. If the message length spans multiple lines, the ProxySG automatically formats the string for multiline capability.



The welcome banner text is overridden by the policy property `ftp.welcome_banner()`. This is required for explicit proxy requests, when doing proxy authentication, and also when the policy property `ftp.server_connection(deferred|immediate)` is set to defer the connection.

3. Click **Apply.**

Related CLI Syntax to Define the Default FTP Banner

```
#SGOS#(config) ftp welcome-banner "message"
```

Related CPL Syntax to Create Policy that Overrides the Default Banner

```
<Proxy>
  ftp.welcome_banner("message")
```

If entering text that spans more than one line, use `$(crlf)` for line breaks.

Viewing FTP Statistics

See Chapter 8: "Intercepting and Optimizing HTTP Traffic" on page 111 for information about viewing the FTP statistics.

Chapter 8: Intercepting and Optimizing HTTP Traffic

This chapter describes how to configure the HTTP proxy to manage traffic and accelerate performance in your environment. Whether you use the Web browser to access productivity applications across the WAN or to access content on the Internet, the configure the HTTP proxy to control, secure, and accelerate all traffic that arrives on port 80.

By default, the service on port 80 is set in Bypass mode. To intercept all traffic on port 80, see "To configure the HTTP proxy to intercept traffic:" on page 113.

The HTTP proxy is designed to control Web traffic, providing:

- Security
- Authentication
- Virus Scanning and Patience Pages
- Performance, achieved through Object Caching and Object Pipelining

Before Reading Further

Before reading this chapter you should be familiar with the concepts in the following user guides:

- "About Proxy Listeners" on page 28.
- Section B: "Creating or Editing a Proxy Service" on page 37, for detailed information on configuring proxy services.
- About Authentication Modes in *Volume 4: Securing the Blue Coat ProxySG Appliance*.
- Configuring the Application Delivery Network in *Volume 5: Advanced Networking*, if you want to optimize ADN performance on the HTTP Proxy.

Topics in this Chapter

This chapter includes information about the following topics:

- Section A: "About the HTTP Proxy Service" on page 113
- Section B: "Configuring the HTTP Proxy Performance" on page 119
- Section C: "Caching Authenticated Data (CAD) and Caching Proxy Authenticated Data (CPAD)" on page 141
- Section D: "Viewing HTTP/FTP Statistics" on page 145
- Section E: "Supporting IWA Authentication in an Explicit HTTP Proxy" on page 149

How Do I...?

To navigate this chapter, identify the task to perform and click the link:

How do I...?	See...
Intercept traffic on the HTTP Proxy?	"Changing the HTTP Proxy Service to Intercept All IP Addresses on Port 80" on page 113
Edit an HTTP proxy service? Create a new HTTP Proxy service?	"Creating or Editing an HTTP Proxy Service" on page 114
Configure the HTTP proxy to perform transparent authentication?	See Step 4 in "Creating or Editing an HTTP Proxy Service" on page 114
Elect to enable / disable protocol detection?	Step 4c in "To create or edit an HTTP proxy service:" on page 115
Configure the HTTP Proxy for object freshness?	"Allocating Bandwidth to Refresh Objects in Cache" on page 135 Step 4 in "To set HTTP default object caching policy:" on page 126
Bypass the cache or not cache content using policy?	Refer to: *Volume 6: The Visual Policy Manager and Advanced Policy* *Volume 10: Content Policy Language Guide* Use either the VPM or CPL to create policy that allows for bypassing the cache or for prohibiting caching based on your needs.
Choose a proxy acceleration profile?	"Selecting an HTTP Proxy Acceleration Profile" on page 127
Configure the HTTP proxy to be a: server accelerator or reverse proxy? forward proxy? server-side bandwidth accelerator?	 "Using the Portal Profile" on page 128 "Using the Normal Profile" on page 128 "Using the Bandwidth Gain Profile" on page 128
Fine-tune the HTTP Proxy for bandwidth gain?	"Fine-Tuning Bandwidth Gain" on page 135 "Using Byte-Range Support" on page 136
Configure Internet Explorer to explicitly proxy HTTP traffic?	"Supporting IWA Authentication in an Explicit HTTP Proxy" on page 149

Section A: About the HTTP Proxy Service

A proxy can service requests without contacting the Origin Content Server by retrieving content saved from a previous request made by the same client or another client. This is called caching.

An HTTP proxy caches copies of frequently requested resources on its local hard disk and thereby significantly reduces upstream bandwidth usage and cost, while significantly increasing performance.

Proxy services define the ports and addresses where a ProxySG listens for incoming requests. The ProxySG has two HTTP service listeners defined by default — one for all IPs listening transparently on port 80 and another explicit listener on port 8080. While you can intercept SSL traffic on either port, to enable the ProxySG to detect the presence of SSL traffic you must enable **Detect Protocol** on the service so that the SSL traffic is handed off to the SSL Proxy. For more information on SSL proxy functionality, see "Managing the SSL Proxy" on page 177.

Further, you can create a bypass list on the ProxySG, to exclude the interception of requests sent from specific clients to specific servers and disable caching of the corresponding responses. The static bypass list also turns off all policy control and acceleration for each matching request. For example, for all clients visiting www.bluecoat.com you might exclude interception and caching of all requests, the corresponding responses, acceleration and policy control. If you want to create a static bypass list, used only in a transparent proxy environment, see "About the Bypass List" on page 49.

Configuring the HTTP Proxy Service Options

This section describes how to change the default service options and add new services.

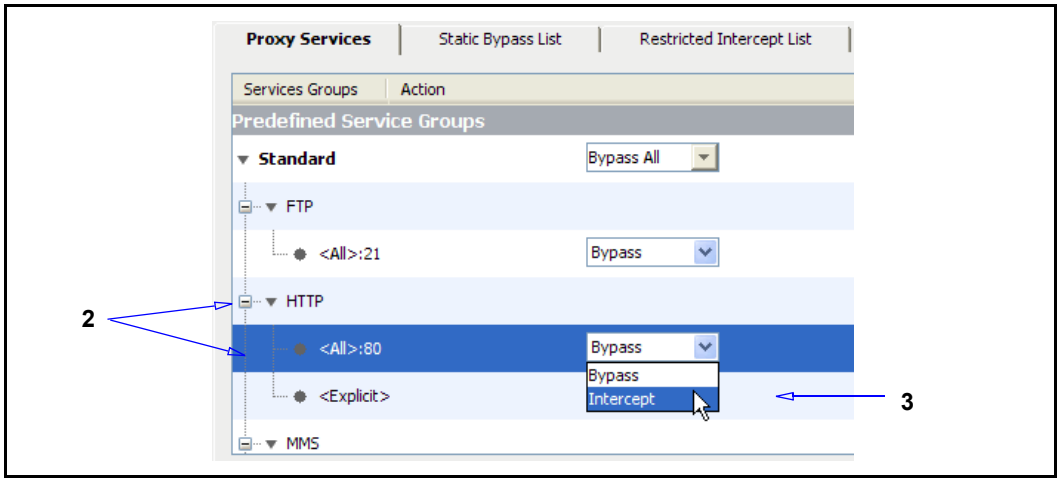
Changing the HTTP Proxy Service to Intercept All IP Addresses on Port 80

By default (upon upgrade and on new systems), the ProxySG has an HTTP proxy service listener configured on port 80. The service is configured to listen to all IP addresses, but is set in Bypass mode.

The following procedure describes how to change the service to Intercept mode.

To configure the HTTP proxy to intercept traffic:

1. From the Management Console, select **Configuration > Services > Proxy Services**.
2. To edit an existing HTTP proxy service, highlight the service and click **Edit**. To create a new proxy service, click **New**.



3. Scroll to Standard service group; click to expand. Scroll to the HTTP service line; click to expand the list of listeners for the HTTP service.
4. Select **Intercept** from the drop-down list for the listener on port 80. Notice that while the default **Action** for the port 80 listener is **Bypass**, the default for the **Explicit** port 8080 listener is **Intercept.**
5. Click **Apply.**

Creating or Editing an HTTP Proxy Service

Although the ProxySG provides a default service, you might create a new HTTP service to cater to an application server in your environment that gates access to its clients using HTTP. You need to provide a name for the corresponding service and add listeners that match on the HTTP traffic for that server's IP address. The benefits of creating a new proxy service are:

- You can view traffic pertaining to this specific service under **Statistics > Traffic Mix** and **Statistics > Traffic History.** Thereby, you can distinguish traffic specific to the server from other HTTP traffic.
- You can write policy that matches on the specific service. This helps in executing discrete and granular decisions on handling this traffic apart from other HTTP traffic.

Use the instructions in this section to create a new HTTP proxy service or to edit an existing one.

The table below lists the attributes that you can edit on the HTTP proxy service.

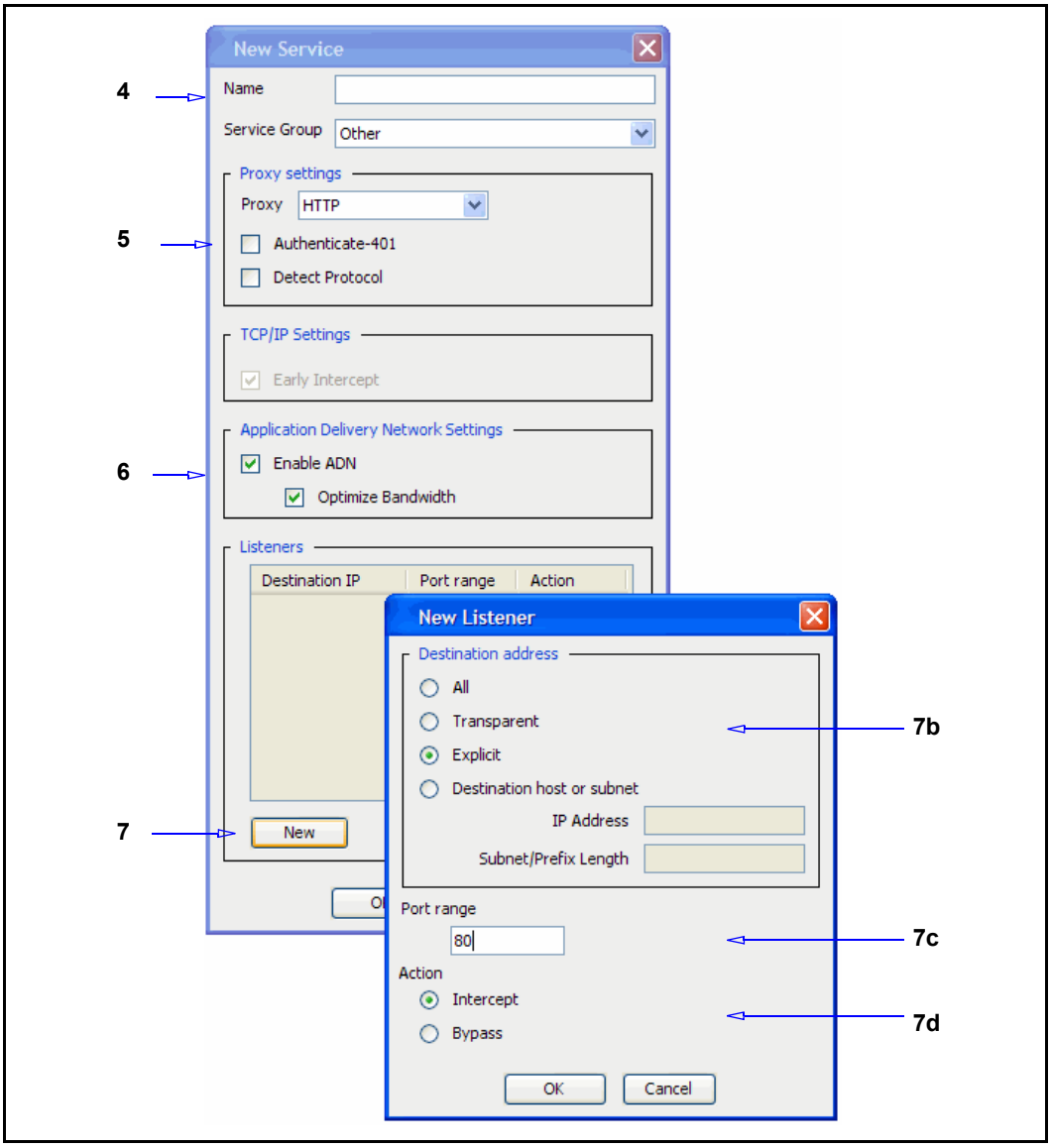
Table 8–1 Customizable Attributes for the HTTP Proxy Service

Attributes	Notes
Authenticate 401	Forces transparent proxy authentication (cookie or IP, depending on the configuration) for all requests received on the port. This is particularly useful in proxy chaining scenarios. Default: Disabled
Detect Protocol	Detects the protocol being used and helps accelerate the flow of traffic. When detect protocol is enabled, the TCP connection must be fully established before the proxy contacts the OCS. Using policy, you can enhance granularity for protocol detection by matching on a richer set of conditions. Default: Disabled
Enable ADN	Secures and accelerates the delivery of applications across the distributed enterprise. Enabling ADN does not guarantee that the connections are accelerated by ADN. For both explicit deployments and transparent deployments, ADN acceleration occurs only when an ADN concentrator is present near the OCS. For Internet traffic, an ADN concentrator must be configured to be an Internet gateway. Default: Enabled
Optimize Bandwidth	Enables byte caching and compression of traffic over the ADN network. When ADN is configured on the ProxySG, this setting helps reduce the amount of bandwidth sent over ADN tunnels. Default: Enabled

To create or edit an HTTP proxy service:

1. Review Table 8–1 on page 115, for planning information and defaults.
2. From the Management Console, select **Configuration > Services > Proxy Services**.
3. To edit an existing HTTP proxy service, highlight the service and click **Edit**. To create a new proxy service, click **New**.

Section A: About the HTTP Proxy Service



4. If you are creating a new HTTP proxy service, enter a meaningful name in the **Name** field.
5. Configure the Proxy Settings options:
   a. Verify HTTP is selected in the drop-down box under **Proxy settings**.
   b. Select the **Authenticate 401** check box if you want all transparent and explicit requests received on the port to always use transparent authentication.
   c. Select the **Detect Protocol** check box to automatically detect the protocol being used.
6. (Optional) Configure ADN options:
   a. **Enable ADN**: Controls whether ADN is enabled for a specific service.

b. **Optimize Bandwidth**: When ADN is enabled, this setting manages byte caching and compression of traffic over the ADN network

7. (Optional when editing an existing service) Add a new listener, specifying the port at which the ProxySG handles the traffic as HTTP. The listener can match on a specific destination IP address or subnet and destination port or port range, and specifies what action to perform on the traffic that matches.

   To add a new listener:

   a. Click **New;** or click **Edit.**

   b. Select a Destination IP address option.

   c. In the **Port Range** field, enter the ports on which the service should listen. The default ports for HTTP are 80 and 8080.

   d. Select the default behavior for the service: **Bypass** tells the service to ignore any traffic. **Intercept** configures the service to intercept the traffic that is being proxied.

   e. Click **OK** to close the dialog.

8. Click **Apply.**

Relevant CLI Syntax to Create/Edit an HTTP Proxy Service:

- To enter configuration mode for the service:

```
SGOS#(config) proxy-services
SGOS#(config proxy-services) create http service-name
SGOS#(config proxy-services) edit service-name
```

- The following subcommands are available:

```
SGOS#(config service-name) add {transparent | explicit | all |
ip_address | ip_address/subnet-mask} {port | first_port-last_port}
[intercept | bypass]
SGOS#(config service-name) attribute {authenticate-401 {enable |
disable} | adn-optimize {enable | disable} | detect-protocol {enable |
disable} | use-adn {enable | disable}}
SGOS#(config service-name) bypass {transparent | explicit | all |
ip_address | ip_address/subnet-mask} {port | first_port-last_port}
SGOS#(config service-name) exit
SGOS#(config service-name) intercept {transparent | explicit | all |
ip_address | ip_address/subnet-mask} {port | first_port-last_port}
SGOS#(config service-name) remove {transparent | explicit | all |
ip_address | ip_address/subnet-mask} {port | first_port-last_port}
SGOS#(config service-name) view
```

Configuring IE for Web FTP with an Explicit HTTP Proxy

Web FTP is used when a client connects in explicit mode using HTTP and accesses an `ftp://` URL. The ProxySG translates the HTTP request into an FTP request for the origin content server (OCS), if the content is not already cached. Further, it translates the FTP response with the file contents into an HTTP response for the client.

Since a Web FTP client uses HTTP to connect to the ProxySG, the HTTP proxy manages this Web FTP traffic. For an explicitly configured HTTP proxy, Internet Explorer version 6.0 users accessing FTP sites over HTTP must disable the **Enable folder view for FTP sites** browser setting.

To disable Web FTP in Internet Explorer v6.0:

1. Select **Tools > Internet Options**.
2. Click the **Advanced** tab.
3. Clear the **Enable folder view for FTP sites** option.
4. Click **OK**.

For information on using FTP, see "Managing the File Transport Protocol (FTP) Proxy" on page 101.

Section B: Configuring the HTTP Proxy Performance

The HTTP proxy alleviates the latency in data retrieval and optimizes the delivery of HTTP traffic through object caching and object pipelining. Caching minimizes the transmission of data over the Internet and over the distributed enterprise, thereby improving bandwidth use. Pipelining allows data to be pre-fetched, even before the client requests it, and caches it to be served immediately upon request. Hence, it directly improves response time.

For objects in cache, the ProxySG's intelligent caching mechanism maintains object freshness. This is achieved by periodically refreshing the contents of the cache, while maintaining the performance within your network.

The method of storing objects on disk is critical for performance and scalability. SGOS, the operating system on the ProxySG, uses an object store system which hashes object lookups based on the entire URL. This hashing allows access to objects with far fewer lookups, as compared to a directory-based filesystem found in traditional operating systems. While file systems run poorly when they are full, a cache achieves its highest performance when it is full.

This section describes the methods you can use to configure the HTTP proxy to optimize performance in your network.

For...	See...
Customizing the object caching policy	"Customizing the HTTP Object Caching Policy" on page 119.
Choosing a proxy acceleration profile that meets your specific needs	"Selecting an HTTP Proxy Acceleration Profile" on page 127.
Fine-tuning bandwidth gain	"Fine-Tuning Bandwidth Gain" on page 135.

Customizing the HTTP Object Caching Policy

Object caching is the saving of an application object locally so that it can be served for future requests without requiring retrieval from the OCS. Objects can, for example, be documents, videos or images on a Web page. When objects are cached, the only traffic that crosses the WAN are permission checks (when required) and verification checks that ensure that the copy of the object in cache is still fresh. By allowing objects to be shared across requests and users, object caching greatly reduces the bandwidth required to retrieve contents and the latency associated with user requests.

For more information on how the ProxySG executes permission checks to ensure authentication over HTTP, see "Caching Authenticated Data (CAD) and Caching Proxy Authenticated Data (CPAD)" on page 141.

In case of a reverse proxy, object caching reduces the load on the OCS and improves scalability of the OCS.

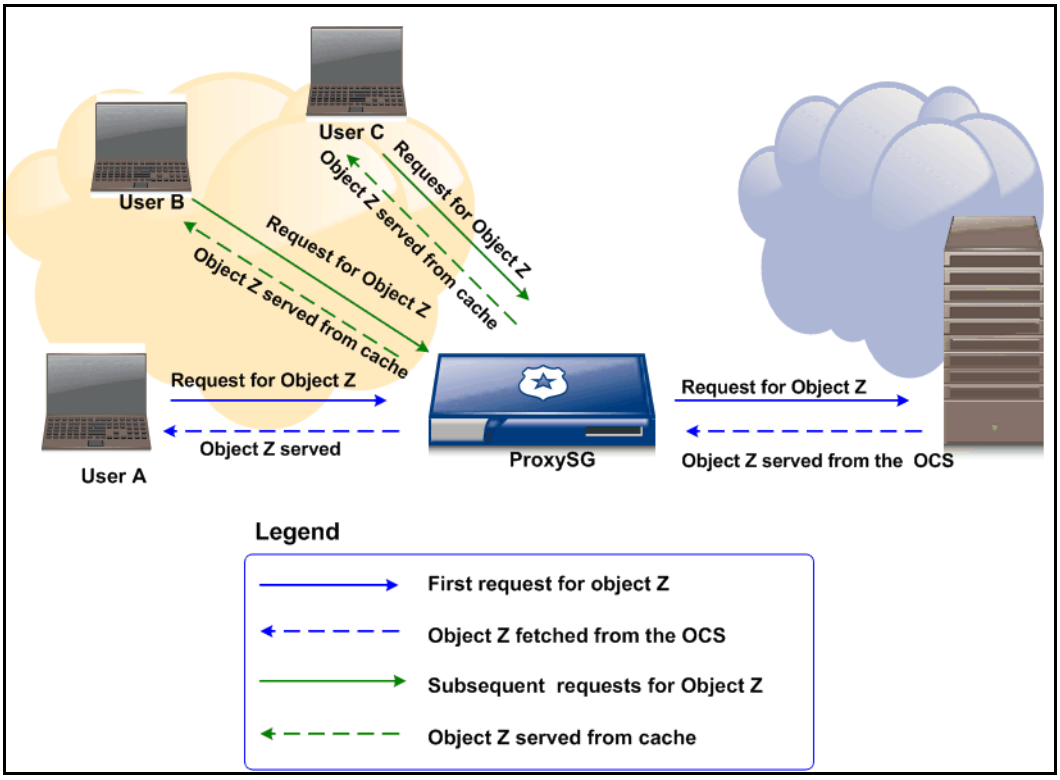


Figure 8–1 Object Caching on the ProxySG

Before you begin customizing your HTTP Proxy policy, read the following concepts:

- "About Object Pipelining" on page 120
- "About HTTP Object Types" on page 121
- "About Meta Tags" on page 122
- "About Tolerant HTTP Request Parsing" on page 122
- "To set HTTP default object caching policy:" on page 126, to configure the global defaults for object caching.

About Object Pipelining

A Web page is typically composed of dozens of objects. When a client requests a Web page, all the objects must be retrieved to display the Web page. This object retrieval process presents a delay for the end user — for example, serial retrieval of the content would create a significant time-lag.

Although modern day browsers open multiple connections with the OCS to retrieve objects in parallel, the ProxySG further accelerates the process with its Object Pipelining algorithm which supports nested pipelines that are up to three levels deep.

The Object Pipelining algorithm allows the ProxySG to open as many simultaneous TCP connections as the origin server allows, and retrieves objects in parallel. The proxy also pre-fetches objects based on pipelined requests. If for example, a pipelined HTML object has other embedded objects, the HTTP proxy will pre-fetch those embedded objects from the Web server without a request from the client. The objects are then ready to be delivered from the cache straight to the user, as fast as the client can request them.

While object pipelining enhances the user experience by minimizing latency and improving response times for first-time Web page requests, it could increase bandwidth utilization. Therefore by default, to avoid an increase in bandwidth utilization, object pipelining is disabled for the reverse proxy and bandwidth gain profiles. It is enabled, by default, only on the forward proxy — Normal profile, where enhancing the response time for clients is vital.

About HTTP Object Types

HTTP proxy categorizes HTTP objects into three types:

- Type-T: The OCS specifies explicit expiration time.
- Type-M: Expiration time is not specified; however, the last modified time is specified by the OCS.
- Type-N: Neither expiration nor last modified time has been specified.

The SGOS maintains the freshness for all three types of cached HTTP objects using the Asynchronous Adaptive Refresh (AAR) algorithm. Using the AAR algorithm the ProxySG performs *freshness checks* with the OCS to expunge old content from cache and to replace it with updated content. This technique significantly speeds subsequent requests for the same object by removing latency as the objects in cache is refreshed asynchronous to actual user requests.

To maximize the freshness of the next access to objects in the cache, asynchronous revalidations are performed on those objects based on their relative popularity and the amount of time remaining before their estimated time of expiration.

Estimated expiration times vary as object content changes are observed during such asynchronous revalidations. This happens even for type-T objects because the expiration times of type-T objects are not always perfectly managed by Webmasters of content servers. However, for situations where such management can be trusted, you can configure the proxy to reduce speculative revalidation of type-T objects. The terms revalidation and refresh mean are used synonymously—to assess the freshness of an object by sending a conditional GET request to the object's OCS. Table 8–4 on page 129 lists the components that you can configure for type-T objects.

On the ProxySG, object pipelining improves response times for first-time Web page requests, and the AAR algorithm significantly speeds subsequent requests by removing the latency involved in refreshing the objects.

About Meta Tags

A meta tag is a hidden tag that placed in the `<head>` of an HTML document. It provides descriptions and keywords for search engines and can contain the attributes — `content`, `http-equiv`, and `name`. Meta tags with an `http-equiv` attribute are equivalent to HTTP headers.

The ProxySG does not parse HTTP meta tag headers if:

- The meta tag does not appear within the first 256 bytes of the HTTP object body. To be parsed, relevant HTTP meta tags must appear within the first 256 bytes of the HTTP object body.
- The ProxyAV that is connected to your ProxySG, adds or modifies the meta tags in its response to the ProxySG. The response body modified by the ProxyAV is not parsed.

Planning Considerations

You can use CPL properties in the `<Cache>` layer to control meta tag processing. The CPL commands can be used in lieu of the check boxes for parsing meta tags through the Management Console. For details on the meta-tags, see Step 7 in "To set HTTP default object caching policy:" on page 126.

The following CPL commands are applicable for HTTP proxy, HTTP refresh, and HTTP pipeline transactions:

```
http.response.parse_meta_tag.Cache-Control(yes|no)
http.response.parse_meta_tag.Expires(yes|no)
http.response.parse_meta_tag.Pragma.no-cache(yes|no)
```

VPM support to control the processing of meta tags is not available.

Related CLI Syntax to Parse Meta Tags

```
SGOS#(config) http [no] parse meta-tag cache-control
SGOS#(config) http [no] parse meta-tag expires
SGOS#(config) http [no] parse meta-tag pragma-no-cache
```

About Tolerant HTTP Request Parsing

The tolerant HTTP request parsing flag causes certain types of malformed requests to be processed instead of being rejected.The defaults are:

- Proxy Edition: The HTTP tolerant request parsing flag is not set, by default, The ProxySG blocks malformed HTTP requests, returning a *400 Invalid Request* error.
- MACH5 Edition: The HTTP tolerant request parsing flag is set by default. Malformed HTTP requests are not blocked.

Implementation of HTTP Tolerant Request Parsing

By default, a header line that does not begin with a <Tab> or space character must consist of a header name (which contains no <Tab> or space characters), followed by a colon and an optional value.

When the tolerant HTTP request parsing flag is either not set or is disabled, if the header name and required details are missing, the ProxySG blocks malformed HTTP requests and returns a *400 Invalid Request* error.

With tolerant request parsing enabled, a request header name is allowed to contain <Tab> or space characters, and if the request header line does not contain a colon, then the entire line is taken as the header name.

A header containing only one or more <Tab> or space characters is considered ambiguous. The ProxySG cannot discern if this is a blank continuation line or if it is a blank line that signals the end of the header section. By default, an ambiguous blank line is illegal, and an error is reported. With tolerant request parsing enabled, an ambiguous blank line is treated as the blank line that ends the header section.

To enable the HTTP tolerant request parsing flag:

Note: This feature is only available through the CLI.

From the `(config)` prompt, enter the following command to enable tolerant HTTP request parsing (the default is disabled):

```
SGOS#(config) http tolerant-request-parsing
```

To disable HTTP tolerant request parsing:

```
SGOS#(config) http no tolerant-request-parsing
```

Configuring the Global Defaults on the HTTP Object Caching Policy

The ProxySG offers multiple configuration options that allow you to treat cached objects in a way that best suits your business model.

The following table lists the options that you can configure.

Table 8–2 Settings for Configuring the Object Caching Policy

Settings to Configure Object Caching	Notes
Setting the maximum object cache size	Determines the maximum object size to store in the ProxySG. All objects retrieved that are greater than the maximum size are delivered to the client but are not stored in the ProxySG. Default: 1024 MB

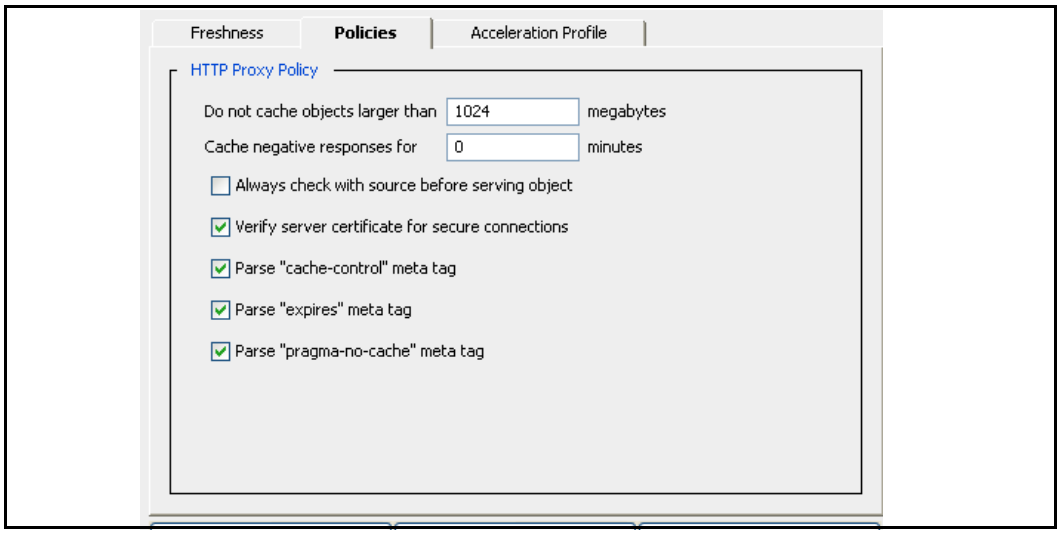
Settings to Configure Object Caching	Notes
Setting the TTL for negative responses in cache	Determines the number of minutes the SGOS stores negative responses for requests that could not be served to the client. The OCS might send a client error code (4xx response) or a server error code (5xx response) as a response to some requests. If you configure the ProxySG to cache negative responses for a specified number of minutes, it returns the negative response in subsequent requests for the same page or image for the specified length of time. The ProxySG will not attempt to fetch the request from the OCS. Therefore, while server-side bandwidth is saved, you could receive negative responses to requests that might otherwise have been served by accessing the OCS. By default, the ProxySG does not cache negative responses. It always attempts to retrieve the object from the OCS, if it is not already in cache. Default: 0 minutes
Forcing freshness validation before serving an object from cache	Verifies that each object is fresh upon access. Enabling this setting has a significant impact on performance because the HTTP proxy revalidates requested cached objects with the OCS before serving them to the client. This results in a negative impact on bandwidth gain. Therefore, do not enable this configuration unless absolutely required. For enabling, select the **Always check with source before serving object** check box. Default: Disabled
Forcing HTTPS server certificate validation	Always verifies the certificate of the OCS for HTTPS connections. For enabling, select the **Verify server certificate for secure connections.** Default: Disabled

Settings to Configure Object Caching	Notes
Parsing HTTP meta tag headers	Determines how HTTP meta tag headers are parsed in the HTML documents. The meta tags that can be enabled for parsing are: • **Cache-control meta tag** The sub-headers that are parsed when this check box is selected are: `private, no-store, no-cache, max-age, s-maxage, must-revalidate, proxy-revalidate` • **Expires meta tag** This directive parses for the date and time after which the document should be considered expired. • **Pragma-no-cache meta tag** This directive indicates that cached information should not be used and instead requests should be forwarded to the OCS. Default: Disabled
Allocating bandwidth on the HTTP proxy for maintaining freshness of the objects in cache	Allows you to specify a limit to the amount of bandwidth the ProxySG uses to achieve the desired freshness. Blue Coat recommends letting the ProxySG manage bandwidth allocation, the default setting. For more information see, "Allocating Bandwidth to Refresh Objects in Cache" on page 135. Default: **Let the SG appliance manage refresh bandwidth.**

The above settings serve as defaults on the proxy. If you want a more granular caching policy, for example— setting the TTL for an object, use Blue Coat Content Policy Language (CPL). You can also use the VPM or CPL to bypass the cache or to prohibit caching for a specific domain or server. Refer to *Volume 10: Content Policy Language Guide* for more information.

To set HTTP default object caching policy:

1. Review Table 8–2, for planning information and defaults.
2. From the Management Console, select **Configuration > Proxy Settings > HTTP Proxy > Policies.**



3. Set the maximum object cache size. In the **Do not cache objects larger than** field, enter the maximum object size to cache. The default is 1024 MB.
4. Set the negative response Time-to-Live. In the **Cache negative responses for** field, enter the number of minutes SGOS stores negative responses. The default is 0.
5. Force freshness validation. To always verify that each object is fresh upon access, select the **Always check with source before serving object** check box. Enabling this setting has a significant impact on performance, do not enable this configuration unless absolutely required.
6. Force HTTPS server certificate validation. If you communicate with an origin content server (OCS) through HTTPS and want the OCS certificate to be verified, be sure that **Verify server certificate for secure connections** is selected.
7. Disable meta-tag parsing. The default is to parse HTTP meta tag headers in HTML documents if the MIME type of the object is text/html.

   To disable meta-tag parsing, clear the check box for:

   - **Parse cache-control meta tag**
     The following sub-headers are parsed when this check box is selected: `private, no-store, no-cache, max-age, s-maxage, must-revalidate, proxy-revalidate.`
   - **Parse expires meta tag**
     This directive parses for the date and time after which the document should be considered expired.

- **Parse pragma-no-cache meta tag**
  This directive indicates that cached information should not be used and instead requests should be forwarded to the OCS.

8. Click **OK**; click **Apply.**

Related CLI Syntax to Set HTTP Proxy Default Policy

- To enter configuration mode:

```
SGOS#(config)caching
SGOS#(config caching)
```

- The following subcommands are available:

```
SGOS#(config caching) always-verify-source
SGOS#(config caching) no always-verify-source
SGOS#(config caching) max-cache-size megabytes
SGOS#(config caching) negative-response minutes
SGOS#(config caching) refresh automatic
SGOS#(config caching) refresh no automatic
SGOS#(config caching) refresh bandwidth kbps
SGOS#(config) http parse meta-tag {cache-control | expires | pragma-
no-cache}
SGOS#(config)http no parse meta-tag
```

See Also

- "Customizing the HTTP Object Caching Policy" on page 119.
- Clearing the Object Cache in *Volume 9: Managing the Blue Coat ProxySG Appliance*
- "Selecting an HTTP Proxy Acceleration Profile" on page 127.

Selecting an HTTP Proxy Acceleration Profile

A proxy profile offers a collection of attributes that determine object caching and object pipelining behavior. The attributes are pre-selected to meet a specific objective — reduce response time for clients, reduce load on the OCS, reduce server-side bandwidth usage.

Based on your needs, you can select any of the three profiles offered or you can create a customized profile by selecting or clearing the options available within a profile.

The available proxy profile are:

- Normal (the default setting) acts as a client accelerator, and is used for enterprise deployments.
- Portal acts as a server accelerator (reverse proxy), and is used for Web hosting.
- Bandwidth Gain is used for Internet Service Provider (ISP) deployments.

Using the Normal Profile

Normal is the default profile and can be used wherever the ProxySG is used as a normal forward proxy. This profile is typically used in enterprise environments, where the freshness of objects is more important than controlling the use of server-side bandwidth. The Normal profile is the profile that most follows the HTTP standards concerning object revalidation and staleness. Additionally, prefetching (pipelining) of embedded objects and redirects is enabled, which reduces response time for clients.

Using the Portal Profile

When configured as a server accelerator or reverse proxy, the ProxySG improves object response time to client requests, scalability of the origin content server (OCS) site, and overall Web performance at the OCS. A server accelerator services requests meant for an OCS, as if it is the OCS itself.

Using the Bandwidth Gain Profile

The Bandwidth-Gain profile is useful wherever server-side bandwidth is an important resource. This profile is typically used in Internet Service Provider (ISP) deployments. In such deployments, minimizing server-side bandwidth is most important. Therefore, maintaining the freshness of an object in cache is less important than controlling the use of server-side bandwidth. The Bandwidth-Gain profile enables various HTTP configurations that can increase page response times and the likelihood that stale objects are served, but it reduces the amount of server-side bandwidth required.

The table below shows the configuration for each profile.

Table 8–3 Normal, Portal, and Bandwidth Gain Profiles

Configuration	Normal Profile	Portal Profile	Bandwidth Gain
Pipeline embedded objects in client requests	Enabled	Disabled	Disabled
Pipeline embedded objects in prefetch requests	Enabled	Disabled	Disabled
Pipeline redirects for client requests	Enabled	Disabled	Disabled
Pipeline redirects for prefetch requests	Enabled	Disabled	Disabled
Cache expired objects	Enabled	Disabled	Enabled
Bandwidth Gain Mode	Disabled	Disabled	Enabled
Substitute GET for IMS (if modified since)	Disabled	Enabled	Enabled
Substitute GET for PNC (Pragma no cache)	Disabled	Enabled	Disabled
Substitute GET for HTTP 1.1 conditionals	Disabled	Enabled	Enabled
Substitute GET for IE (Internet Explorer) reload	Disabled	Enabled	Disabled

Table 8–3 Normal, Portal, and Bandwidth Gain Profiles (Continued)

Configuration	Normal Profile	Portal Profile	Bandwidth Gain
Never refresh before expiration	Disabled	Enabled	Enabled
Never serve after expiration	Disabled	Enabled	Disabled

When a ProxySG is first manufactured, it is set to a *Normal* profile. Depending on your needs, you can use the *Bandwidth Gain* profile or the *Portal* profile. You can also combine elements of all three profiles, as needed for your environment.

About HTTP Proxy Profile Configuration Components

The table below gives a definition of the customizable HTTP proxy profile settings. Both the Management Console field and CLI `(config)` command text is included.

Table 8–4 Description of Profile Configuration Components

Management Console Check box Field	CLI `(config)` Command	Definition
Pipeline embedded objects in client request	`http [no] pipeline client requests`	This configuration item applies only to HTML responses. When this setting is enabled, and the object associated with an embedded object reference in the HTML is not already cached, HTTP proxy acquires the object's content before the client requests the object. This improves response time dramatically. If this setting is disabled, HTTP proxy does not acquire embedded objects until the client requests them.
Pipeline redirects for client request	`http [no] pipeline client redirects`	When this setting is enabled, and the response of a client request is one of the redirection responses (such as 301, 302, or 307 HTTP response code), then HTTP proxy pipelines the object specified by the `Location` header of that response, provided that the redirection location is an HTML object. This feature improves response time for redirected URLs. If this setting is disabled, HTTP proxy does not pipeline redirect responses resulting from client requests.

Section B: Configuring the HTTP Proxy Performance

Table 8–4 Description of Profile Configuration Components (Continued)

Management Console Check box Field	CLI (config) Command	Definition
Pipeline embedded objects in prefetch request	`http [no] pipeline prefetch requests`	This configuration item applies only to HTML responses resulting from pipelined objects. When this setting is enabled, and a pipelined object's content is also an HTML object, and that HTML object has embedded objects, then HTTP proxy also pipelines those embedded objects. This nested pipelining behavior can occur three levels deep at most. If this setting is disabled, the HTTP proxy does not perform nested pipelining.
Pipeline redirects for prefetch request	`http [no] pipeline prefetch redirects`	When this setting is enabled, HTTP proxy pipelines the object specified by a redirect location returned by a pipelined response. If this setting is disabled, HTTP proxy does not try to pipeline redirect locations resulting from a pipelined response.
Substitute Get for IMS	`http [no] substitute if-modified-since`	If the time specified by the `If-Modified-Since:` header in the client's conditional request is greater than the last modified time of the object in the cache, it is a strong indication that the copy in the cache is stale. If so, HTTP proxy does a conditional GET to the OCS, based on the last modified time of the cached object. To control this aspect of the SGOS treatment of the `If-Modified-Since:` header, enable the Substitute Get for IMS setting. When this setting is enabled, a client time condition greater than the last modified time of the object in the cache does not trigger revalidation of the object. Note: All objects do not have a last-modified time specified by the OCS.

Table 8–4 Description of Profile Configuration Components (Continued)

Management Console Check box Field	CLI (config) Command	Definition
Substitute Get for HTTP 1.1 conditionals	`http [no] substitute conditional`	HTTP 1.1 provides additional controls to the client over the behavior of caches concerning the staleness of the object. Depending on various `Cache-Control:` headers, the ProxySG can be forced to consult the OCS before serving the object from the cache. For more information about the behavior of various `Cache-Control:` header values, refer to RFC 2616. If the Substitute Get for HTTP 1.1 Conditionals setting is enabled, HTTP proxy ignores the following `Cache-Control:` conditions from the client request: • `"max-stale" [ "=" delta-seconds ]` • `"max-age" "=" delta-seconds` • `"min-fresh" "=" delta-seconds` • `"must-revalidate"` • `"proxy-revalidate"`
Substitute Get for PNC	`http [no] substitute pragma-no-cache`	Typically, if a client sends an HTTP GET request with a `Pragma: no-cache` or `Cache-Control: no-cache` header (for convenience, both are hereby referred to as PNC), a cache must consult the OCS before serving the content. This means that HTTP proxy always re-fetches the entire object from the OCS, even if the cached copy of the object is fresh. Because of this, PNC requests can degrade proxy performance and increase server-side bandwidth utilization. However, if the Substitute Get for PNC setting is enabled, then the PNC header from the client request is ignored (HTTP proxy treats the request as if the PNC header is not present at all).
Substitute Get for IE reload	`http [no] substitute ie-reload`	Some versions of Internet Explorer issue the `Accept: */*` header instead of the `Pragma: no-cache` header when you click **Refresh**. When an `Accept` header has only the `*/*` value, HTTP proxy treats it as a PNC header if it is a type-N object. You can control this behavior of HTTP proxy with the Substitute GET for IE Reload setting. When this setting is enabled, the HTTP proxy ignores the PNC interpretation of the `Accept: */*` header.

Table 8–4 Description of Profile Configuration Components (Continued)

Management Console Check box Field	CLI (config) Command	Definition
Never refresh before expiration	`http [no] strict-expiration refresh`	Applies only to cached type-T objects. For information on HTTP object types, see "About HTTP Object Types" on page 121. When this setting is enabled, SGOS does not asynchronously revalidate such objects before their specified expiration time. When this setting is disabled, such objects, if they have sufficient relative popularity, can be asynchronously revalidated and can, after a sufficient number of observations of changes, have their estimates of expiration time adjusted accordingly.
Never serve after expiration	`http [no] strict-expiration serve`	Applies only to cached type-T objects. If this setting is enabled, an object is synchronously revalidated before being served to a client, if the client accesses the object after its expiration time. If this setting is disabled, the object is served to the client and, depending on its relative popularity, may be asynchronously revalidated before it is accessed again.
Cache expired objects	`http [no] cache expired`	Applies only to type-T objects. When this setting is enabled, type-T objects that are already expired at the time of acquisition is cached (if all other conditions make the object cacheable). When this setting is disabled, already expired type-T objects become non-cacheable at the time of acquisition.

Table 8–4 Description of Profile Configuration Components (Continued)

Management Console Check box Field	CLI (config) Command	Definition
Enable Bandwidth Gain Mode	`bandwidth-gain {disable | enable}`	This setting controls both HTTP-object acquisition after client-side abandonment and AAR (asynchronous adaptive refresh) revalidation frequency. • HTTP-Object Acquisition When Bandwidth Gain mode is enabled, if a client requesting a given object abandons its request, then HTTP proxy immediately abandons the acquisition of the object from the OCS, if such an acquisition is still in progress. When bandwidth gain mode is disabled, the HTTP proxy continues to acquire the object from the OCS for possible future requests for that object. • AAR Revalidation Frequency Under enabled bandwidth gain mode, objects that are asynchronously refreshable are revalidated at most twice during their estimated time of freshness. With bandwidth gain mode disabled, they are revalidated at most three times. Not all asynchronously refreshable objects are guaranteed to be revalidated.

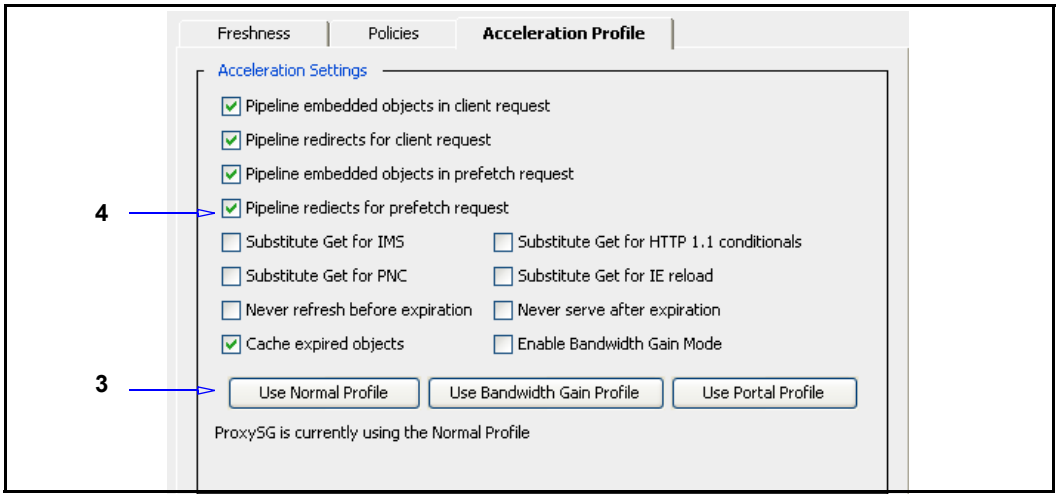
Configuring the HTTP Proxy Profile

You can configure the profile using any of the components discussed above.

To configure the HTTP proxy profile:

1. Review the description of the components for each profile, see Table 8–4 on page 129.
2. From the Management Console, select **Configuration > Proxy Settings > HTTP Proxy > Acceleration Profile**.

   The Acceleration Profile tab displays (Normal is the default profile). Text appears at the bottom of this tab indicating which profile is selected. If you have a customized profile, this text does not appear.



Important: If you have a customized profile and you click one of the **Use Profile** buttons, no record of your customized settings remains. However, once the ProxySG is set to a specific profile, the profile is maintained in the event the ProxySG is upgraded.

3. To select a profile, click one of the three profile buttons (**Use Normal Profile**, **Use Bandwidth Gain Profile**, or **Use Portal Profile**).

   The text at the bottom of the **Acceleration Profile** tab changes to reflect the new profile.

   Note: You can customize the settings, no matter which profile button you select.

4. (Optional) To customize the profile settings, select or clear any of the check boxes (see Table 8–4, "Description of Profile Configuration Components" on page 129 for information about each setting).
5. Click **OK**; click **Apply**.

Related CLI Syntax to Configure the HTTP Proxy Profile

```
SGOS#(config) profile {normal | portal | bwgain}
```

See Also

- "Selecting an HTTP Proxy Acceleration Profile" on page 127.
- "About HTTP Proxy Profile Configuration Components" on page 129.
- "About HTTP Object Types" on page 121.
- "Fine-Tuning Bandwidth Gain" on page 135.

Fine-Tuning Bandwidth Gain

In addition to the components related to top-level profiles, other configurable items affect bandwidth gain. You can set the top-level profile (see "Selecting an HTTP Proxy Acceleration Profile" on page 127) and adjust the following configuration items to fine tune the ProxySG for your environment:

- Allocating bandwidth to refresh objects in cache
- Using Byte-range support
- Enabling the Revalidate pragma-no-cache(PNC)

Allocating Bandwidth to Refresh Objects in Cache

The ProxySG uses as much bandwidth as necessary for refreshing content on frequently accessed cached objects and for maintaining their freshness. The amount of bandwidth used varies depending on client demands.

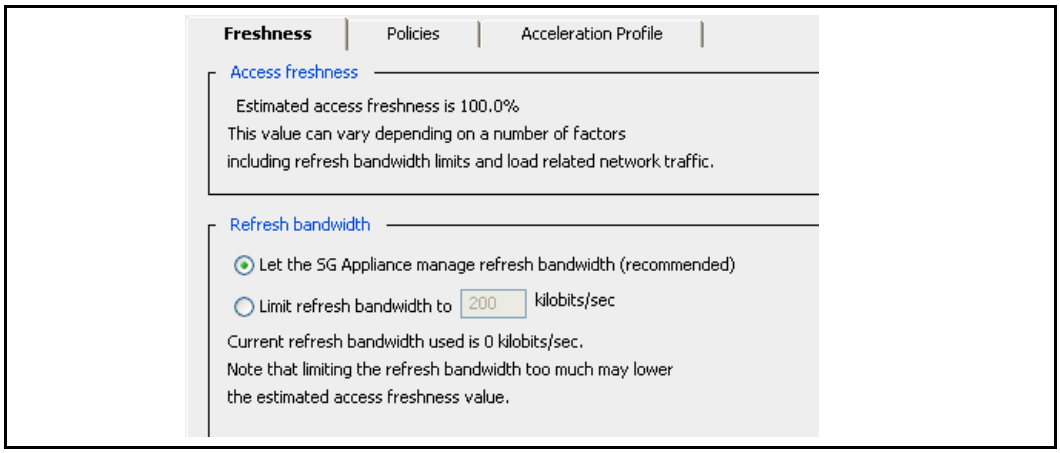
The **Refresh bandwidth** option refers to server-side bandwidth used for all forms of asynchronous refresh activity. If you determine that the ProxySG is using too much bandwidth, you can specify a limit to the amount of bandwidth the ProxySG uses to achieve the desired freshness. Before making adjustments, review the logged statistics and examine the current bandwidth used as displayed in the **Refresh bandwidth** field. It is not unusual for bandwidth usage to spike occasionally, depending on access patterns at the time.

To limit the refresh bandwidth to a specified amount, you must disable automatic management of the bandwidth and explicitly set a bandwidth limit. Setting the refresh bandwidth amount too low can lower the estimated freshness of objects in the cache. If you set the refresh bandwidth amount to zero, the ProxySG does not do active refresh at all.

If the refresh bandwidth configuration remains at the recommended default—**Let the SG appliance manage refresh bandwidth (recommended)** in the Management Console or `SGOS#(config caching) refresh automatic` in the CLI—then the appliance uses whatever bandwidth is available in its efforts to maintain 99.9% estimated freshness of the next access.

To set refresh bandwidth:

1. From the Management Console, select **Configuration > Proxy Settings > HTTP Proxy > Freshness.**



The **Refresh bandwidth** field displays the refresh bandwidth options. The default setting is to allow the ProxySG to manage refresh bandwidth automatically.

Important: Blue Coat strongly recommends that you not change the setting from the default.

2. Do one of the following:
   - To turn off automatic bandwidth refresh, select **Limit refresh bandwidth to** (not recommended). Enter a new value into the **kilobits/sec** field, if necessary.
   - To return the appliance to automatic bandwidth refresh, select **Let the SG Appliance manage refresh bandwidth (recommended).**
3. Click **OK**; click **Apply.**

Relevant CLI Syntax to Set Refresh Bandwidth

```
SGOS#(config) caching
```

- The following subcommands are available:

```
SGOS#(config caching) refresh automatic
SGOS#(config caching) refresh bandwidth kbps
```

Using Byte-Range Support

Byte-range support is an HTTP feature that allows a client to use the `Range:` HTTP header for requesting a portion of an object rather than the whole object. The HTTP proxy supports byte-range support and it is enabled by default.

When Byte-Range Support is Disabled

If byte-range support is disabled, HTTP treats all byte-range requests as non-cacheable. Such requests are never served from the cache, even if the object exists in the cache. The client's request is sent unaltered to the OCS and the response is not cached. Thus, a byte-range request has no effect on the cache if byte-range support is disabled.

When Byte-Range Support is Enabled

If the object is already in cache, the ProxySG serves the byte-range request from the cache itself. However, if the client's request contains a PNC header, the ProxySG always bypasses the cache and serves the request from the OCS.

If the object is not in cache, the ProxySG always attempts to minimize delay for the client.

- If the byte-range requested is near the beginning of the object, that is the start byte of the request is within 0 to 14336 bytes, then the ProxySG fetches the entire object from the OCS and caches it. However, the client is served the requested byte-range only.
- If the byte-range requested is not near the beginning of the object, that is the start byte of the request is greater than 14336 bytes, then the ProxySG fetches only the requested byte-range from the OCS, and serves it to the client. The response is not cached.

Note: The HTTP proxy never caches partial objects, even if byte-range support is enabled.

Since the ProxySG never caches partial objects, bandwidth gain is significantly affected when byte-range requests are used heavily. If, for example, several clients request an object where the start byte offset is greater than 14336 bytes, the object is never cached. The ProxySG fetches the same object from the OCS for each client, thereby causing negative bandwidth gain.

Further, download managers like NetAnts® typically use byte-range requests with PNC headers. To improve bandwidth gain by serving such requests from cache, enable the **revalidate pragma-no-cache** option along with byte-range support. See "Enabling Revalidate Pragma-No-Cache" on page 138.

To configure byte-range support:

Note: Enabling or disabling byte-range support can only be configured through the CLI.

To enable or disable byte-range support, enter one of the following commands at the `(config)` command prompt:

```
SGOS#(config) http byte-ranges
-or-
SGOS#(config) http no byte-ranges
```

Enabling Revalidate Pragma-No-Cache

The pragma-no-cache (PNC) header in a client's request causes the HTTP proxy to re-fetch the entire object from the OCS, even if the cached copy of the object is fresh. This roundtrip for PNC requests can degrade proxy performance and increase server-side bandwidth utilization.

While the **Substitute Get for PNC** configuration completely ignores PNC in client requests and potentially serves stale content, the `revalidate-pragma-no-cache` setting allows you to selectively implement PNC.

When the `revalidate-pragma-no-cache` setting is enabled, a client's non-conditional PNC-GET request results in a conditional GET request sent to the OCS if the object is already in cache. The `revalidate-pragma-no-cache` request allows the OCS to return the **304 Not Modified** response, if the content in cache is still fresh. Thereby, the server-side bandwidth consumed is lesser as the full content is not retrieved again from the OCS.

By default, the revalidate PNC configuration is disabled and is not affected by changes in the top-level profile. When the **Substitute Get for PNC** configuration is enabled (see Table 8–4, "Description of Profile Configuration Components" on page 129 for details), the revalidate PNC configuration has no effect.

To configure the revalidate PNC setting:

Note: The revalidate pragma-no-cache setting can only be configured through the CLI.

To enable or disable the revalidate PNC setting, enter one of the following commands at the `(config)` command prompt:

```
SGOS#(config) http revalidate-pragma-no-cache
-or-
SGOS#(config) http no revalidate-pragma-no-cache
```

Interpreting Negative Bandwidth Gain Statistics

Bandwidth gain represents the overall bandwidth benefit achieved by object and byte caching, compression, protocol optimization, and object caching. Occasionally, you might notice negative bandwidth gain when using the bandwidth gain profile. This negative bandwidth gain is observed because the

client-side cumulative bytes of traffic is lower than the server-side cumulative bytes of traffic for a given period of time. It is represented as a unit-less multiplication factor and is computed by the ratio:

```
client bytes / server bytes
```

Some factors that contribute to negative bandwidth gain are:

- Abandoned downloads (delete_on_abandonment (no))

  When a client cancels a download, the ProxySG continues to download the requested file to cache it for future requests. Since the client has cancelled the download, server-side traffic persists while the client-side traffic is halted. This continued flow of traffic on the server-side causes negative bandwidth gain.

  Further with (delete_on_abandonment (yes)), when a client cancels a download, the ProxySG terminates the connection and stops sending traffic to the client. However, the server may have sent additional traffic to the ProxySG before it received the TCP RESET from the ProxySG. This surplus also causes negative bandwidth gain.

- Refreshing of the cache

  Bandwidth used to refresh contents in the cache contributes to server-side traffic. Since this traffic is not sent to the client until requested, it might cause negative bandwidth gain.

- Byte-range downloads

  When download managers use an open-ended byte-range, such as `Range: bytes 10000-`, and reset the connection after downloading the requested byte-range. The packets received by the ProxySG from the server are greater than those served to the client, causing negative bandwidth gain.

- Download of uncompressed content

  If the ProxySG downloads uncompressed content, but compresses it before serving the content to the client, server-side traffic will be greater than client-side traffic. This scenario is typical in a reverse proxy deployment, where the server offloads the task of gzipping the content to the ProxySG.

- Reduced client-side throughput

  In the short term, you will notice negative bandwidth gain if the client-side throughput is lower than the server-side throughput. If, for example, the ProxySG takes 5 minutes to download a 100 Mb file and takes 10 minutes to serve the file to the client. The ProxySG will reflect negative bandwidth gain for the first 5 minutes.

To view bandwidth usage and bandwidth gain statistics on the HTTP proxy, click **Statistics > Traffic History** tab. Select the HTTP proxy service to view statistics over the last hour, day, week, month, and year. Refer to the Statistics chapter in *Volume 9: Managing the Blue Coat ProxySG Appliance*, for information on the graphs.

Related CLI Syntax to Configure HTTP

The following commands allow you to manage settings for an HTTP proxy.

Use the command below to enter the configuration mode.

```
SGOS# conf t
```

- The following subcommands are available:

```
SGOS#(config) http [no] add-header client-ip
SGOS#(config) http [no] add-header front-end-https
SGOS#(config) http [no] add-header via
SGOS#(config) http [no] add-header x-forwarded-for
SGOS#(config) http [no] byte-ranges
SGOS#(config) http [no] cache authenticated-data
SGOS#(config) http [no] cache expired
SGOS#(config) http [no] cache personal-pages
SGOS#(config) http [no] force-ntlm
SGOS#(config) http ftp-proxy-url root-dir
SGOS#(config) http ftp-proxy-url user-dir
SGOS#(config) http [no] parse meta-tag {cache-control | expires |
pragma-no-cache}
SGOS#(config) http [no] persistent client
SGOS#(config) http [no] persistent server
SGOS#(config) http [no] persistent-timeout client num_seconds
SGOS#(config) http [no] persistent-timeout server num_seconds
SGOS#(config) http [no] pipeline client {requests | redirects}
SGOS#(config) http [no] pipeline prefetch {requests | redirects}
SGOS#(config) http [no] proprietary-headers bluecoat
SGOS#(config) http receive-timeout client num_seconds
SGOS#(config) http receive-timeout refresh num_seconds
SGOS#(config) http receive-timeout server num_seconds
SGOS#(config) http [no] revalidate-pragma-no-cache
SGOS#(config) http [no] strict-expiration refresh
SGOS#(config) http [no] strict-expiration serve
SGOS#(config) http [no] strip-from-header
SGOS#(config) http [no] substitute conditional
SGOS#(config) http [no] substitute ie-reload
SGOS#(config) http [no] substitute if-modified-since
SGOS#(config) http [no] substitute pragma-no-cache
SGOS#(config) http [no] tolerant-request-parsing
SGOS#(config) http upload-with-pasv disable
SGOS#(config) http upload-with-pasv enable
SGOS#(config) http version {1.0 | 1.1}
SGOS#(config) http [no] www-redirect
SGOS#(config) http [no] xp-rewrite-redirect
```

Note: For detailed information about using these commands, refer to *Volume 11: Command Line Interface Reference*.

Section C: Caching Authenticated Data (CAD) and Caching Proxy Authenticated Data (CPAD)

This section describes how the ProxySG caches authenticated content over HTTP. Authentication over HTTP allows a user to prove their identity to a server or an upstream proxy to gain access to a resource.

The ProxySG uses CAD and CPAD to facilitate object caching at the edge and to help validate user credentials. Object caching in the ProxySG allows for lesser bandwidth usage and faster response times between the client and the server or proxy.

The deployment of the ProxySG determines whether it performs CAD or CPAD:

- When the Origin Content Server (OCS) performs authentication, the ProxySG performs CAD.
- When the upstream HTTP Proxy performs authentication, the downstream HTTP proxy or ProxySG executes CPAD.

About Caching Authenticated Data (CAD)

In the CAD scenario, when a user requests a resource that needs authentication, the OCS sends an HTTP 401 error response to the user. The HTTP 401 response also contains information on the authentication schemes that the OCS supports. To prove their identity to the OCS, the user resubmits the initial request along with the authentication details.

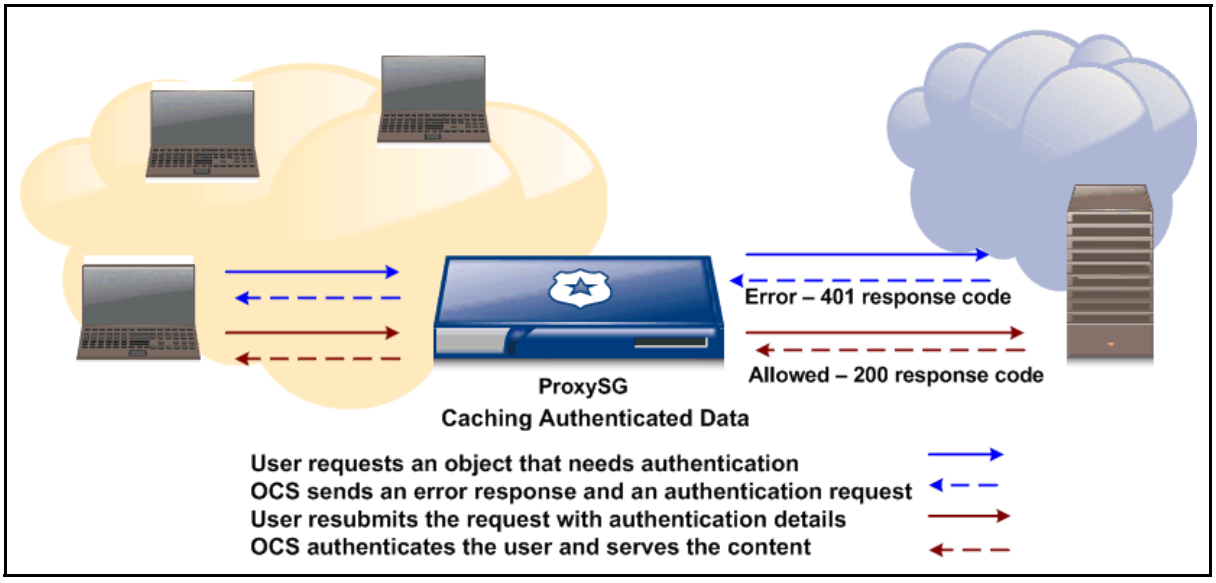


Figure 8–2 CAD: 200 response from the Origin Content Server.

The OCS then sends back one of the following responses:

- HTTP 200 response status, authentication is accepted. The user receives the requested resource.

- HTTP 403 response status, user is not allowed to view the requested resource. The user is authenticated but is not authorized to receive the content, hence the user receives an error message. See Figure 8-2.

When another user accesses the same URL, the ProxySG authenticates the user with the OCS and verifies the freshness of the content using the `Get If Modified Since` request. If the user is authorized and the content has not been modified, the OCS returns an HTTP 304 response message to the ProxySG. The ProxySG then serves the content from cache.

If the content has been modified, the OCS returns the HTTP 200 response along with the modified content.

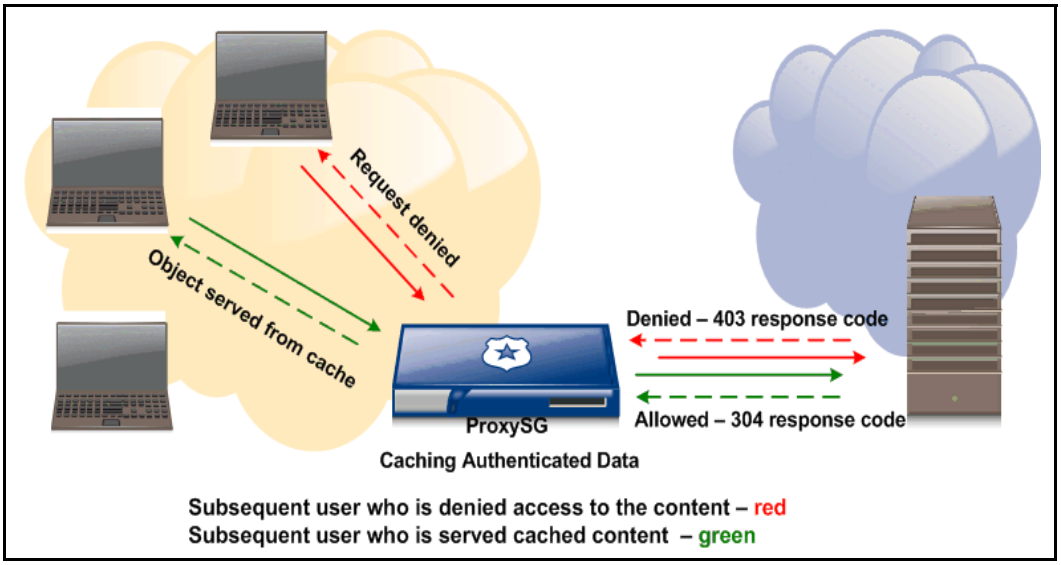


Figure 8–3 CAD: 403 and 304 response codes from the OCS

Note: CAD is applicable only for pure HTTP authentication — the ProxySG caches authenticated data only when the OCS includes the `www-Authenticate` response code in the 401 response header. If, for example, the client accesses an OCS that uses forms-based authentication, the ProxySG does not perform CAD.

About Caching Proxy Authenticated Data (CPAD)

The CPAD deployment uses two ProxySG appliances — a local proxy and a gateway proxy. Figure 8–4 on page 143 below depicts the ProxySG appliances in a CPAD deployment.

When the user requests a resource, **ProxySG1** forwards the request to **ProxySG2**. **ProxySG2** issues the authentication challenge back to the user (a 407 response instead of the 401 response that the OCS serves). Upon successful authentication, **ProxySG2** forwards the request to the OCS and the resource is served to the user.

Section C: Caching Authenticated Data (CAD) and Caching Proxy Authenticated Data (CPAD)

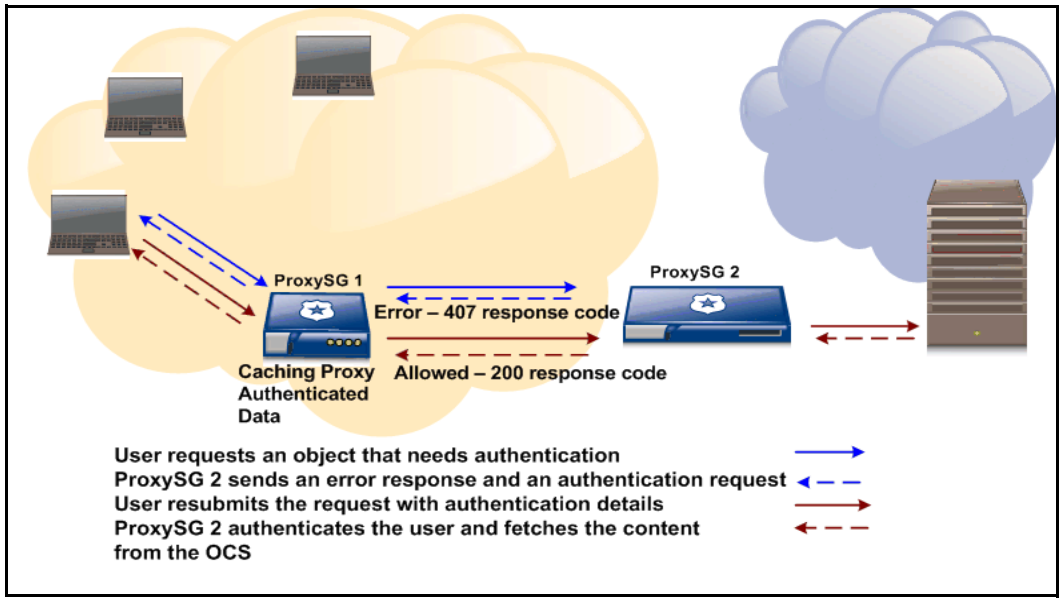


Figure 8–4 CPAD: 200 response from ProxySG 2

In Figure 8–5, **ProxySG1** caches proxy authenticated data and **ProxySG2** performs authentication (instead of the OCS).

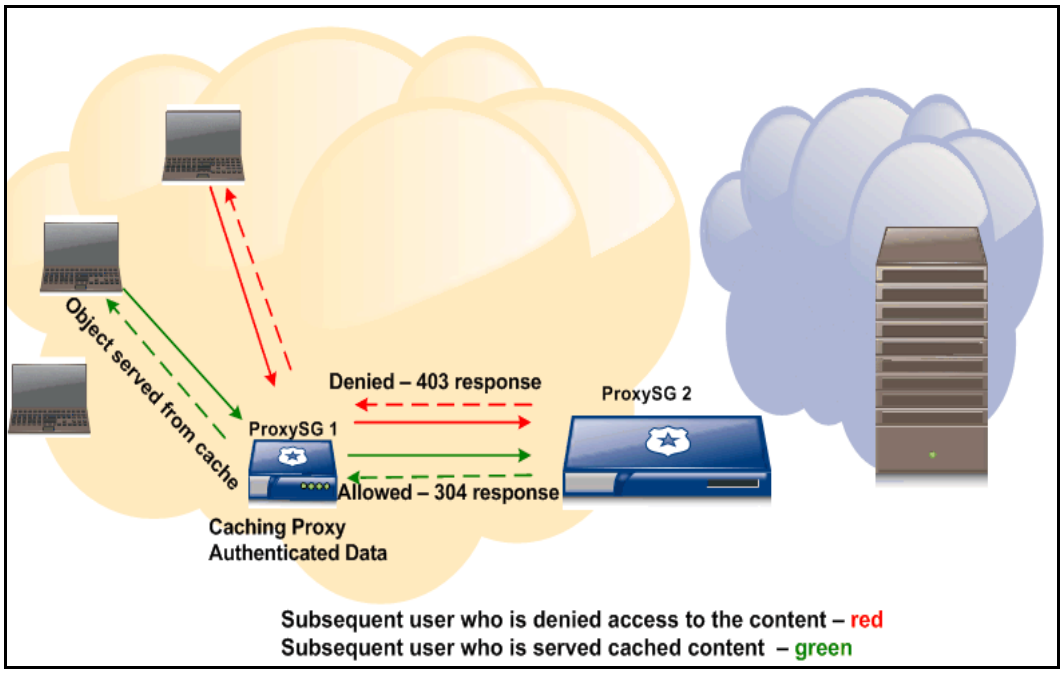


Figure 8–5 CPAD: 407 and 304 responses in a CPAD deployment

For subsequent users who access the same URL, see Figure 8-4, **ProxySG1** forwards all requests to **ProxySG2** with the `Get If Modified Since` request.

ProxySG2 issues the authentication challenge and provides one of the following responses:

- HTTP 200 response status, the user is allowed access to the requested resource but the content has changed.
- HTTP 304 response status, the user is authorized and the content can be served from the cache.
- HTTP 403 response status, the user is not authorized to view the requested resource.
- HTTP 407 response status, the user provided invalid credentials.

Section D: Viewing HTTP/FTP Statistics

HTTP/FTP History Statistics

The **HTTP/FTP History** tabs display bar graphs that illustrate the last 60 minutes, 24 hours, and 30 days for the number of objects served, bytes served, active clients, and client and server compression gain statistics associated with the HTTP, HTTPS, and FTP protocols. The overall client and server compression-gain statistics are displayed under **System Usage**.

Note: You can view current HTTP statistics through the CLI using the `show http-stats` command.

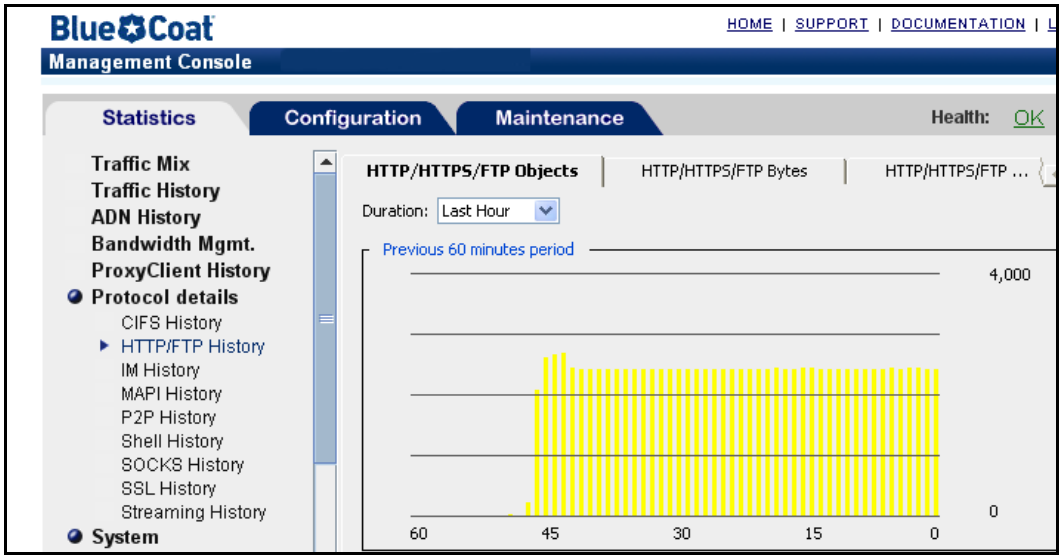
Viewing the Number of HTTP/HTTPS/FTP Objects Served

The HTTP / HTTPS / FTP Objects tab illustrates the device activity over the last 60 minutes, 24 hours, and 30 days. These charts illustrate the total number of objects served from either the cache or from the Web.

The maximum number of objects that can be stored on a ProxySG is affected by a number of factors, including the SGOS version it is running and the hardware platform series.

To view the number of HTTP/HTTPS/FTP objects served:

1. From the Management Console, select **Statistics > Protocol Details > HTTP/FTP History > HTTP/HTTPS/FTP Objects**.
2. Select the **Duration:** from the drop-down list.



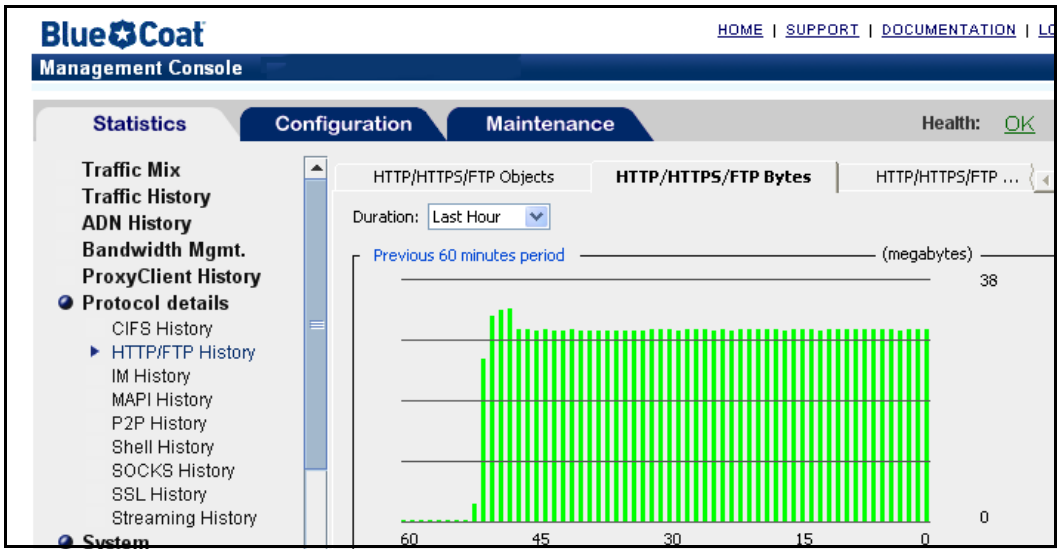
3. (Optional) To set the graph scale to a different value, select a value from the **Graph scale should** drop-down list.

Viewing the Number of HTTP/HTTPS/FTP Bytes Served

The Bytes tab shows the sum total of the number of bytes served from the device over the last 60 minutes, 24 hours, and 30 days. The chart shows the total number of bytes for objects served by the device, including both cache hits and cache misses.

To view the number of HTTP/HTTPS/FTP bytes served:

1. From the Management Console, select **Statistics > Protocol Details > HTTP/FTP History > HTTP/HTTPS/FTP Bytes**.
2. Select the **Duration:** from the drop-down list.



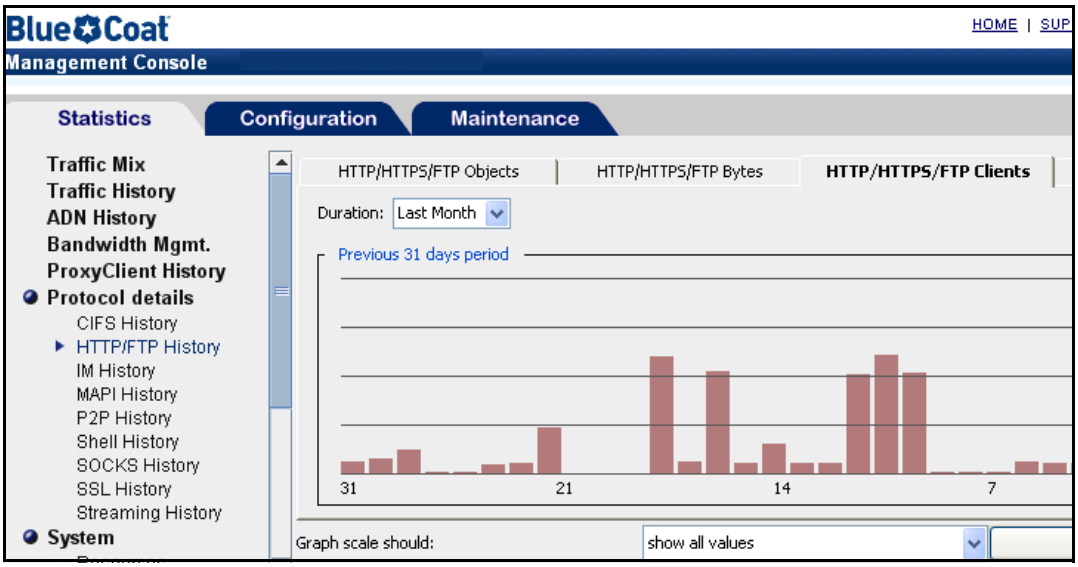
3. (Optional) To set the graph scale to a different value, select a value from the **Graph scale should** drop-down list.

Viewing Active Client Connections

The HTTP / HTTPS / FTP Clients tab shows the maximum number of clients with requests processed over the last 60 minutes, 24 hours, and 30 days. This does not include idle client connections (connections that are open but that have not made a request). These charts allow you to monitor the maximum number of active clients accessing the ProxySG at any one time. In conjunction with the HTTP / HTTPS / FTP Objects and HTTP / HTTPS / FTP Bytes tabs, you can determine the number of clients supported based on load, or load requirements for your site based on a specific number of clients.

To view the number of active clients:

1. From the Management Console select **Statistics > Protocol Details > HTTP/FTP History > HTTP/HTTPS/FTP Clients**.
2. Select the **Duration:** from the drop-down list.



3. (Optional) To set the graph scale to a different value, select a value from the **Graph scale should** drop-down list.

Viewing HTTP/HTTPS/FTP Client and Server Compression Gain Statistics

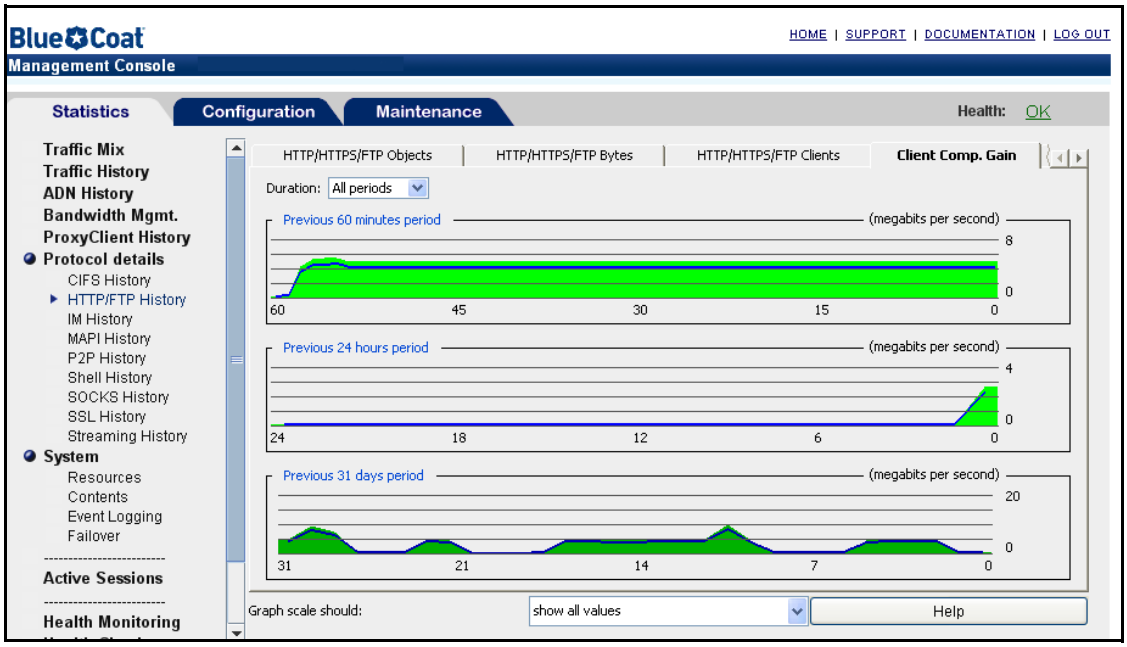
Under HTTP/FTP History, you can view HTTP/FTP client and server compression-gain statistics for the ProxySG over the last 60 minutes, 24 hours, and 30 days in the Client Comp. Gain and the Server Comp. Gain tabs. Overall client and server compression-gain statistics are displayed under System Usage. These statistics are not available through the CLI.

The green display on the bar graph represents uncompressed data; the blue display represents compressed data. Hover your cursor over the graph to see the compressed gain data.

To view HTTP/FTP client compressed gain statistics:

1. From the Management Console, select **Statistics > Protocol Details > HTTP/FTP History > Client Comp. Gain.**
2. Select the **Duration:** from the drop-down list.

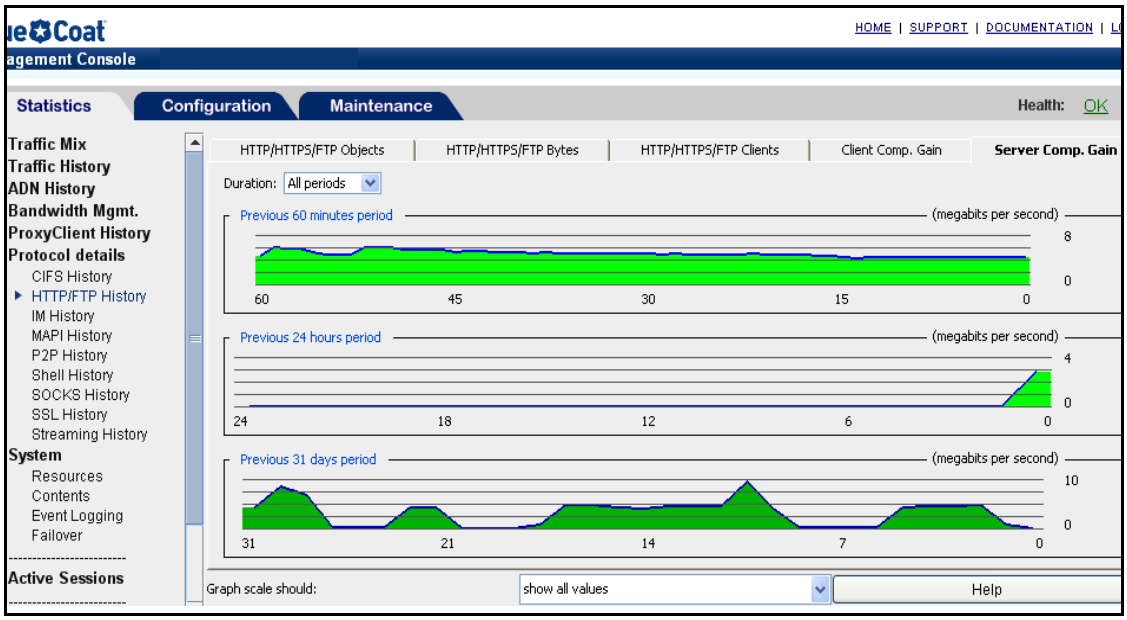
Section D: Viewing HTTP/FTP Statistics



3. (Optional) To set the graph scale to a different value, select a value from the **Graph scale should** drop-down list.

To view HTTP/FTP server compressed gain statistics:

1. From the Management Console, select **Statistics > Protocol Details > HTTP/FTP History > Server Comp. Gain.**
2. Select the **Duration:** from the drop-down list.



3. (Optional) To set the graph scale to a different value, select a value from the **Graph scale should** drop-down list.

Section E: Supporting IWA Authentication in an Explicit HTTP Proxy

Internet Explorer does not allow IWA authentication through a ProxySG when explicitly proxied. To facilitate this authentication, Blue Coat added a `Proxy-Support: Session-based-authentication` header. By default, when the ProxySG receives a 401 authentication challenge from upstream, it sends the `Proxy-Support: Session-based-authentication` header in response.

The Proxy-Support header is not supported if:

- you are using an older browser (Refer to the Release Notes for supported browser versions).
- both the ProxySG and the OCS perform IWA authentication.

In either case, Blue Coat recommends that you disable the header and enable **Force IWA for Server Authentication.** To enable this setting through policy, you must create a Web Access Layer in the Visual Policy Manager, see *Volume 6: The Visual Policy Manager and Advanced Policy*. The **Force IWA for Server Authentication** action, converts the 401-type server authentication challenge to a 407-type proxy authentication challenge that Internet Explorer supports. The ProxySG also converts the resulting Proxy-Authentication headers in client requests to standard server authorization headers, which allows an IWA authentication challenge to pass through when Internet Explorer is explicitly proxied through the appliance.

Disabling the Proxy-Support Header

The Proxy-Support header is sent by default when an explicitly configured ProxySG receives a 401 authentication challenge from upstream.

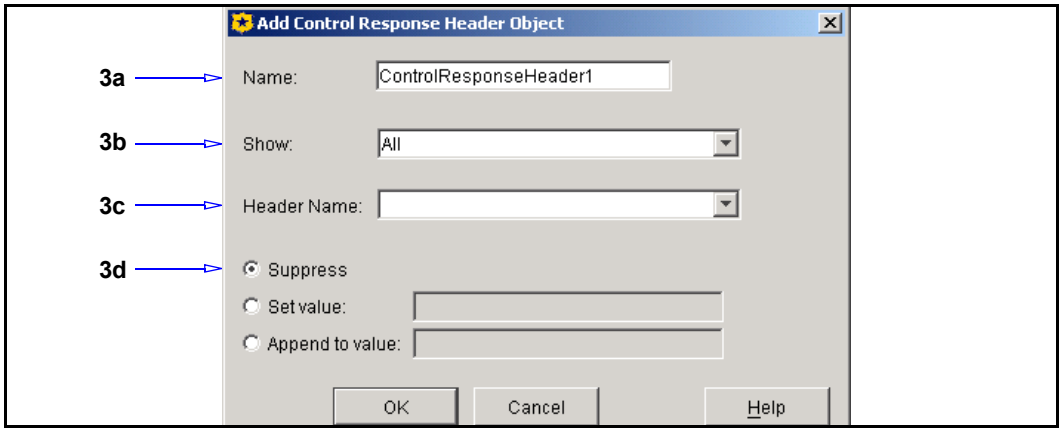
The header modification policy allows you to suppress or modify the Proxy-Support custom header, and prevents the ProxySG from sending this default header. Use either the Visual Policy Manager (VPM) or CPL to disable the header through policy. For complete information on using VPM, refer to *Volume 6: The Visual Policy Manager and Advanced Policy*.

Note: To suppress the Proxy-Support header globally, use the `http force-ntlm` command to change the option. To suppress the header only in certain situations, continue with the procedures below.

To suppress the proxy-support header through VPM:

To suppress the header using VPM, create a new Web Access Layer. Then:

1. Right click in the **Action** field to see the drop-down list; select **Set.**
2. Click **New** to see the drop-down list; select **Control Response Header.**



3. Fill in the fields as follows:
   a. **Name**: Enter a meaningful name.
   b. **Show**: Select **Custom** from the drop-down list.
   c. **Header Name**: Enter Proxy-Support.
   d. Verify **Suppress** is selected.
4. Click **OK**; click **Apply.**

To suppress the proxy-support header through CPL:

Use CPL to define the Proxy-Support custom header object and to specify what action to take. The example below uses Proxy-Support as the action name, but you can choose any name meaningful to you. The result of this action is to suppress the Proxy-Support header

```
<Proxy>
  action.Proxy-Support(yes)

define action Proxy-Support
  delete(response.x_header.Proxy-Support)
end action Proxy-Support
```

Chapter 9: Configuing and Managing an HTTPS Reverse Proxy Service

This chapter describes the Blue Coat HTTPS Reverse Proxy implementation, which:

- Combines hardware-based SSL acceleration with full caching functionality.
- Establishes and services incoming SSL sessions.
- Provides SSL v2.0, SSL v3.0, and TLSv1 protocol support.

Creating an HTTPS reverse proxy is unlike other proxies in that a number of preliminary steps are required before you can use the proxy.

Preliminary steps include:

- Creating or importing a keyring. (Refer to *Volume 4: Securing the Blue Coat ProxySG Appliance* for information on creating or importing a keyring.)
- (If necessary) Creating Certificate Signing Requests (CSRs) that can be sent to Certificate Signing Authorities (CAs).
- Importing server certificates issued by CA authorities for external use and associate them with the keyring. (Refer to *Volume 4: Securing the Blue Coat ProxySG Appliance*.)

  -or-

- Creating certificates for internal use and associate them with the keyring.
- (Optional, if using server certificates from CAs) Importing Certificate Revocation Lists (CRLs) so the ProxySG can verify that certificates are still valid.

When these steps are complete, you can configure the HTTPS Reverse Proxy service.

A common scenario in using HTTPS Reverse Proxy, which connects the client to the ProxySG, is in conjunction with HTTPS *origination*, which is used to connect the to the origin content server (OCS). For more information on this option, see Section B: "Configuring HTTP or HTTPS Origination to the Origin Content Server" on page 156.

Topics in this Chapter

This chapter includes information about the following topics:

- "Configuring the HTTPS Reverse Proxy" on page 152
- "Configuring HTTP or HTTPS Origination to the Origin Content Server" on page 156

Section A: Configuring the HTTPS Reverse Proxy

This section describes how to change the default service options and add new services.

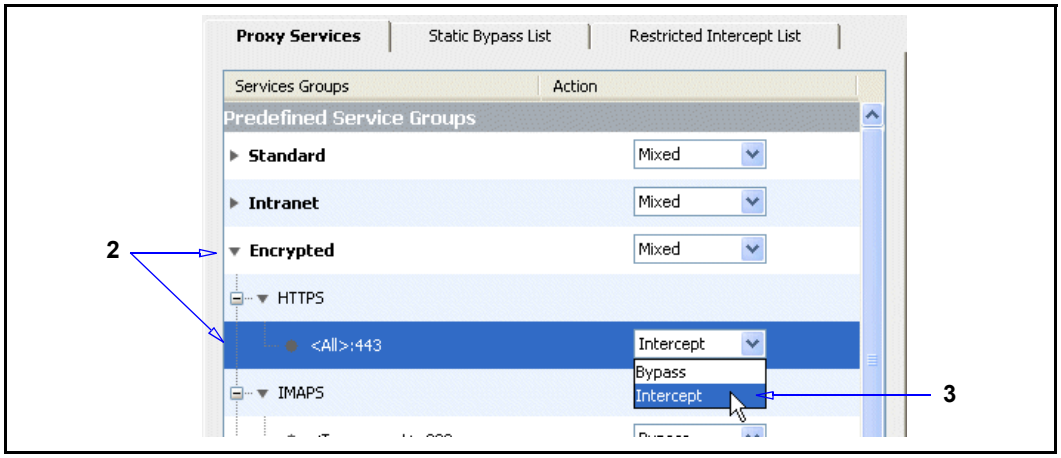
Changing the HTTPS Proxy Service to Intercept All IP Addresses on Port 443

By default (upon upgrade and on new systems), the ProxySG has an HTTPS reverse proxy service configured on port 443. The service is configured to listen to all IP addresses, but is set in Bypass mode.

The following procedure describes how to change the service to Intercept mode.

To configure the HTTPS reverse proxy to intercept traffic:

1. From the Management Console, select **Configuration > Services > Proxy Services.**



2. Scroll the list of service groups and click **Encrypted** to expand the list; select HTTPS to expand the list.
3. From the drop-down list, select **Intercept.**
4. Click **Apply.**

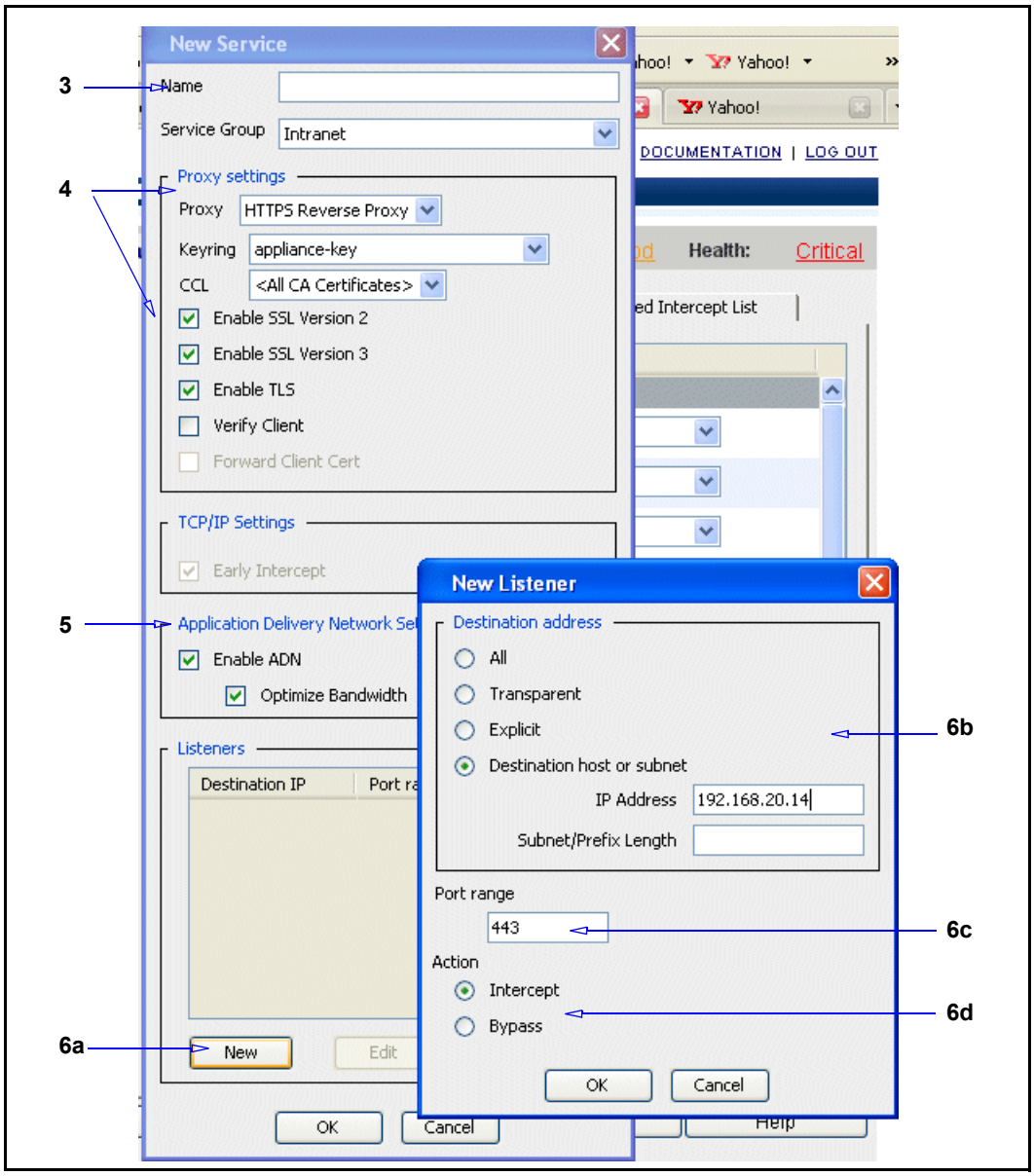
Creating an HTTPS Reverse Proxy Service

This section describes how to create a new HTTPS reverse proxy service.

To create or edit an HTTPS reverse proxy service:

1. Select **Configuration > Services > Proxy Services.**
2. Click **New.**

Section A: Configuring the HTTPS Reverse Proxy



3. Give the proxy has a meaningful name.
4. Configure **Proxy Settings** options:
   a. Verify that **HTTPS Reverse Proxy** is selected in the **Proxy settings** drop-down list.
   b. In the **Keyring** drop-down list, select any already created keyring that is on the system. The system ships with a default keyring that is reusable for each HTTPS service.

Note: The **configuration-passwords-key** keyring that shipped with the ProxySG does *not* contain a certificate.

The **appliance-key** keyring does contain a certificate if you have Internet connectivity, but it cannot be used for purposes other than appliance authentication. For information about appliance authentication, see Chapter 2 of *Volume 5: Advanced Networking*.

---

c. **CA Cert List:** Use the drop-down list to select any already created list that is on the system.

d. **SSL Versions:** Use the drop-down list to select the version to use for this service. The default is **SSL v2/v3** and **TLS v1**.

e. **Verify Client** (Used with the **Forward Client Certificate** option.). Selecting this checkbox enables the **Forward Client Certificate** and puts the extracted client certificate information into the `Client-Cert` header that is included in the request when it is forwarded to the origin content server. The header contains the certificate serial number, subject, validity dates, and issuer (all as `name=value` pairs). The actual certificate itself is not forwarded.

f. **Forward Client Cert**: (Should be used with the **Verify Client** option.) Selecting this option puts the extracted client certificate information into a header that is included in the request when it is forwarded to the OCS.

5. Configure ADN options:

   a. **Enable ADN**: Controls whether ADN is enabled for a specific service. Enabling ADN does not guarantee the connections are accelerated by ADN. The actual enable decision is determined by ADN routing (for explicit deployment) and network setup (for transparent deployment)

   b. The **Optimize Bandwidth** option is selected by default if you enabled ADN optimization during initial configuration. Clear this option if you are not configuring ADN optimization.

6. Add a new listener:

   a. Click **New** to add a new listener to the HTTPS Reverse Proxy; click **Edit** to change the current settings.

   b. Select a Destination IP address option from the drop-down list.

   c. Identify the port to which this service to listens.

   d. Select the default behavior for the service: **Bypass** tells the service to ignore any traffic. **Intercept** configures the service to intercept the traffic that is being proxied.

   e. Click **OK** to close the New Listener dialog.

7. Click **OK** to close the New Service dialog.

8. Click **Apply.**

Relevant CLI Syntax to Create/Edit an HTTPS-Reverse-Proxy Service

- To enter configuration mode for the service:

```
SGOS#(config) proxy-services
SGOS#(config proxy-services) create https-reverse-proxy service-name
SGOS#(config proxy-services) edit service-name
```

- The following subcommands are available:

```
SGOS#(config service-name) add {transparent | explicit | all |
ip_address | ip_address/subnet-mask} {port | first_port-last_port}
[intercept | bypass]
SGOS#(config service-name) attribute {ccl list_name | cipher-suite
cipher-suite | forward-client-cert {enable | disable}| keyring
keyring_id | ssl-versions {sslv2 | sslv3 | tlsv1 | sslv2v3 | sslv2tlsv1
| sslv3tlsv1 | sslv2v3tlsv1} | use-adn {enable | disable}| verify-
client {enable | disable}}
SGOS#(config service-name) bypass {transparent | explicit | all |
ip_address | ip_address/subnet-mask} {port | first_port-last_port}
SGOS#(config service-name) exit
SGOS#(config service-name) intercept {transparent | explicit | all |
ip_address | ip_address/subnet-mask} {port | first_port-last_port}
SGOS#(config service-name) remove {transparent | explicit | all |
ip_address | ip_address/subnet-mask} {port | first_port-last_port}
SGOS#(config service-name) view
```

Section B: Configuring HTTP or HTTPS Origination to the Origin Content Server

In previous procedures, you configured HTTPS Reverse Proxy to the ProxySG. In two common termination scenarios, you must also configure HTTPS origination to the Origin Content Server (OCS).

The first two scenarios are used to provide a secure connection between the proxy and server, if, for example, the proxy is in a branch office and is not co-located with the server.

Figure 9–1 Scenario 1: HTTPS Reverse Proxy with HTTPS Origination

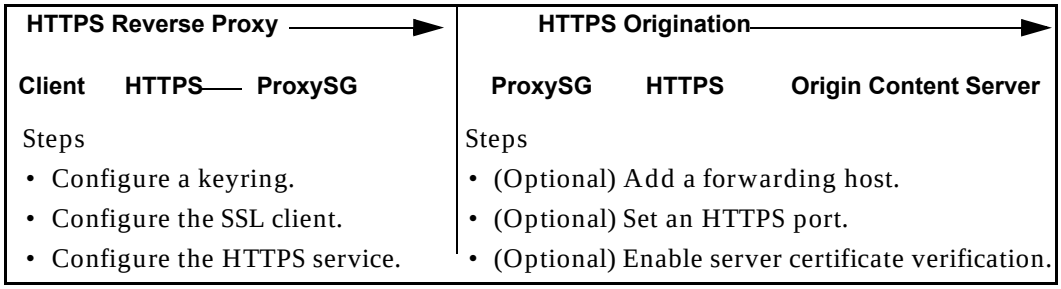
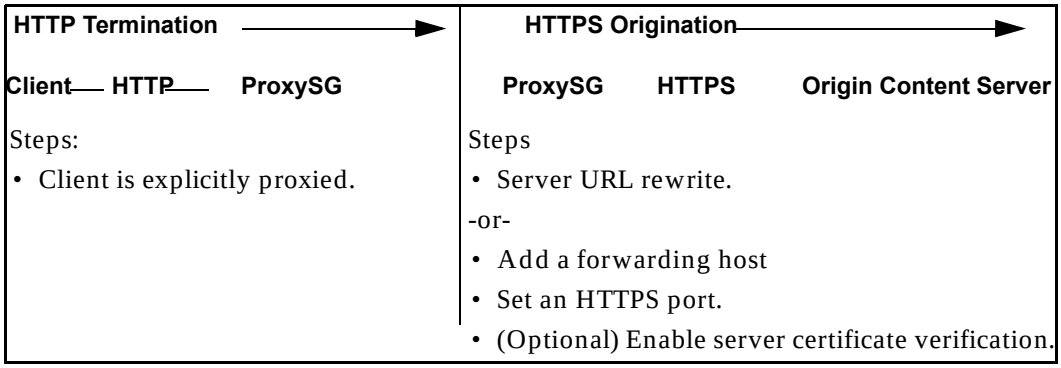


Figure 9–2 Scenario 2: HTTP Termination with HTTPS Origination



Using server URL rewrite is the preferred method. For information on rewriting the server URL, refer to *Volume 10: Content Policy Language Guide*.

To configure HTTPS origination:

At the `(config)` command prompt, enter the following commands:

```
SGOS#(config forwarding) create host_alias hostname
https[=port_number] server ssl-verify-server=yes
```

where:

Table 9–1 HTTPS Origination Commands

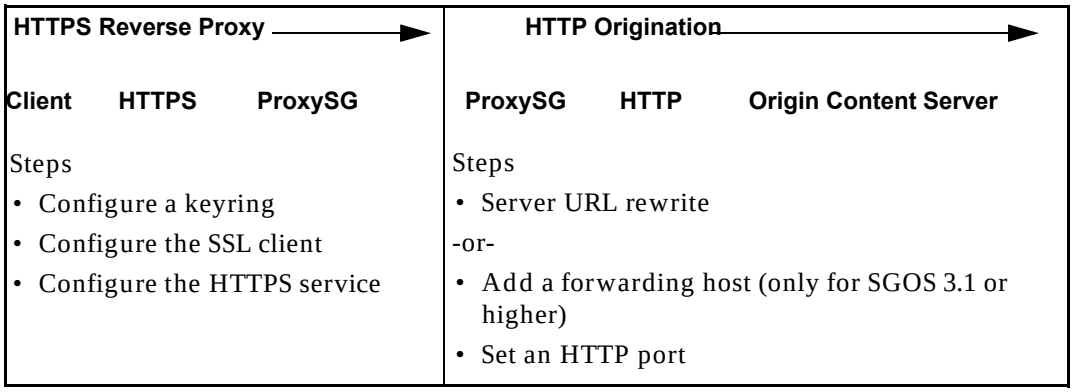
Option	Parameters	Description
host_alias	*alias_name*	Specifies the alias name of the OCS.
host_name		Specifies the hostname or IP address of the OCS, such as www.bluecoat.com.

Table 9–1 HTTPS Origination Commands (Continued)

Option	Parameters	Description
`https`	[=*port_number*]	Specifies the port number on which the OCS is listening.
server		Specifies to use the relative path for URLs in the HTTP header because the next hop is a Web server, not a proxy server. Proxy is the default.
`ssl-verify-server=`	`yes | no`	Specifies whether the upstream server certificate should be verified. You can only enable this command if the upstream host is a server, not a proxy.

The next scenario is useful when the ProxySG is deployed as a reverse proxy. This scenario is used when it's not necessary for a secure connection between the proxy and server. For information on using the ProxySG as a reverse proxy, see "Selecting an HTTP Proxy Acceleration Profile" on page 127.

Figure 9–3 Scenario 3: HTTPS Reverse Proxy with HTTP Origination



Using server URL rewrite is the preferred method. For information on rewriting the server URL, refer to *Volume 10: Content Policy Language Guide*.

You can only configure HTTP origination through the CLI. You cannot use the Management Console.

To configure HTTP origination:

At the `(config)` command prompt, enter the following commands:

```
SGOS#(config forwarding) create host_alias host_name
http[=port_number] server
```

where:

Table 9–2 HTTP Origination Commands

host_alias	*alias_name*	Specifies the alias name of the OCS.
host_name		Specifies the hostname or IP address of the OCS, such as www.bluecoat.com.

Table 9–2 HTTP Origination Commands (Continued)

`http`	`[=port_number]`	Specifies the port number on the OCS in which HTTP is listening.
`server`		server specifies to use the relative path for URLs in the HTTP header because the next hop is a Web server, not a proxy server. Proxy is the default.

Creating Policy for HTTP and HTTPS Origination

Forwarding hosts must be already created on the ProxySG before forwarding policy can be created.

To create a policy using CPL:

```
<forward>
  url.host=host_name forward(host_alias)
```

To create a policy using VPM:

1. In the VPM module, create a Forwarding layer.
2. Set the Destination to be the URL of the OCS.

   Set the Action to forward to the forwarding host and configure parameters to control forwarding behavior.

Chapter 10: Managing Shell Proxies

This chapter discusses how to configure the Telnet shell proxy. Shell proxies provide shells which allow a client to connect to the ProxySG. In this version, only a Telnet shell proxy is supported.

Topics in this Chapter

This chapter includes information about the following topics:

- "About Shell Proxies" on page 159
- "Customizing Policy Settings for Shell Proxies" on page 160
- "About Telnet Shell Proxies" on page 161
- "Configuring the Telnet Shell Proxy Service Options" on page 162
- "Creating or a New Telnet Shell Proxy Service" on page 163
- "Viewing Shell History Statistics" on page 166

About Shell Proxies

Using a shell proxy, you can:

- terminate a Telnet protocol connection either transparently or explicitly.
- authenticate users either transparently or explicitly.
- view the access log.
- enforce policies specified by CPL.
- communicate though an upstream SOCKS gateway and HTTP proxy using the CONNECT method.

Within the shell, you can configure the prompt and various banners using CPL `$substitutions`. You can also use hard-coded text instead of CPL substitutions (available substitutions are listed in the table below). The syntax for a CPL substitution is:

```
$(CPL_property)
```

Table 10–1 CPL Substitutions for Shell Proxies

Substitution	Description
`proxy.name` or `appliance.name`	Configured name of the ProxySG.
`proxy.address`	IP address of the appliance on which this connection is accepted.
`proxy.card`	Adapter number of the appliance on which this connection is accepted.

Table 10–1 CPL Substitutions for Shell Proxies

Substitution	Description
client.protocol	This is telnet.
client.address	IP address of the client.
proxy.primary_address or appliance.primary_address	Primary address of the proxy, not where the user is connected.
release.id	SGOS version.

Customizing Policy Settings for Shell Proxies

To manage a shell proxy through policy, you can use the conditions, properties, and actions listed below. For information on using CPL to manage shell proxies, refer to *Volume 10: Content Policy Language Guide*.

Conditions

- All time and date related triggers
- All exception related triggers
- All server_url triggers
- All url triggers
- All authentication related triggers
- category=
- client.address=
- proxy.address=
- proxy.card=
- proxy.port=
- client.protocol=
- user-defined conditions
- client.protocol=telnet
- url.scheme=telnet

Properties

- allow, deny, force_deny
- action.action_name{yes|no}
- All trace() properties
- All access_log() properties
- All log.xxx() properties
- access_server(yes|no)
- authenticate.force(yes|no)
- authenticate(realm)
- exception(exception_id[, *details*])
- force_exception(exception_id[, *details*])
- forward(alias_list | no)
- forward.fail_open(yes | no)
- reflect_ip(auto | no | client | vip | *ip_address*)
- socks_gateway(alias_list | no)
- socks_gateway.fail_open(yes | no)
- telnet.prompt(no | string)
- telnet.realm_banner(no | string)
- telnet.welcome_banner(no | string)

The banner strings support $-sign substitutions.

Actions

- `rewrite(url.host, host_regex_pattern, replacement_pattern)`
- `rewrite(url, url_regex_pattern, replacement_pattern)`
- `set(url_port, port_number)`
- `log_message()`
- `notify_email(subject, body)`
- `notify_snmp(message)`

Boundary Conditions for Shell Proxies

- A hardcoded timeout of five minutes is enforced from the acceptance of a new connection until destination information is provided using the Telnet command.
- If proxy authentication is enabled, users have three chances to provide correct credentials.
- Users are not authenticated until destination information is provided.
- Users can only enter up to an accumulated 2048 characters while providing the destination information. (Previous attempts count against the total number of characters.)
- Connection to an upstream HTTP proxy is not encouraged.
- If connections from untrustworthy IP address or subnet are not desired, then a client IP / subnet-based *deny* policy must be written.

About Telnet Shell Proxies

The Telnet shell proxy allows you to manage a Telnet protocol connection to the ProxySG. Using the Telnet shell proxy, the ProxySG performs:

- Explicit termination without proxy authentication, where you explicitly connect through Telnet to the ProxySG hostname or IP address. In this case, the ProxySG provides a shell.
- Explicit termination with proxy authentication, where after obtaining the destination host and port information from user, the ProxySG challenges for proxy credentials. After the correct proxy credentials are provided and authenticated, the appliance makes an upstream connection and goes into tunnel mode. In this case, the appliance provides a shell.
- Transparent termination without proxy authentication, where the ProxySG intercepts Telnet traffic through an L4 switch, software bridge, or any other transparent redirection mechanism. From the destination address of TCP socket, the ProxySG obtains OCS contact information and makes the appropriate upstream connection, either directly or through any configured proxy. For more information on configuring a transparent proxy, see Appendix A: "Explicit and Transparent Proxy" on page 213.

- Transparent termination with proxy authentication, where, after intercepting the transparent connection, the ProxySG challenges for proxy credentials. After the correct proxy credentials are provided and authenticated, the ProxySG makes an upstream connection and goes into tunnel mode.

After in the shell, the following commands are available:

- `help`: Displays available commands and their effects.
- `telnet` *`server`*`[:`*`port`*`]`: Makes an outgoing Telnet connection to specified server. The colon (:) between server and port can be replaced with a space, if preferred.
- `exit`: Terminates the shell session.

Configuring the Telnet Shell Proxy Service Options

This section describes how to change the default service options and add new services.

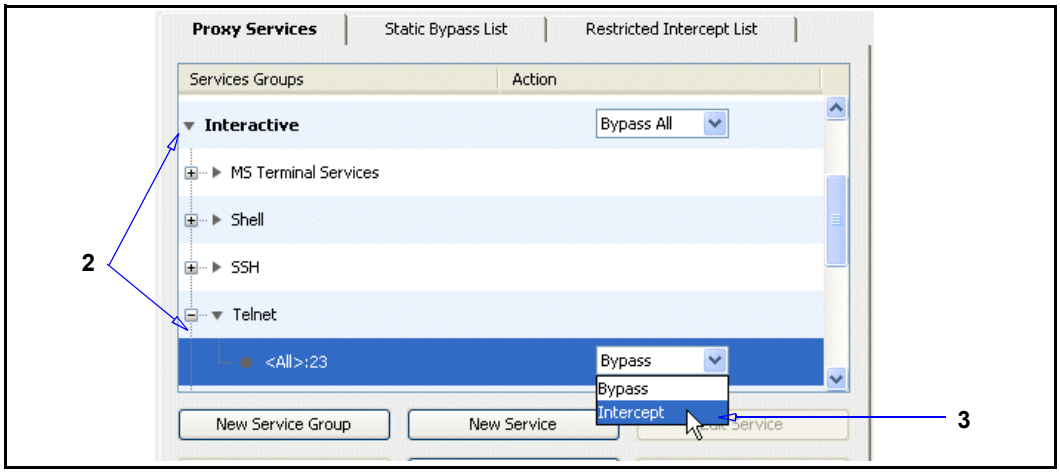
Changing the Telnet Shell Proxy Service to Intercept All IP Addresses on Port 23

The service is configured to listen to all IP addresses, but is set in Bypass mode. The following procedure describes how to change the service to Intercept mode. Default settings are:

- Proxy Edition–a Telnet proxy service is configured but disabled on port 23 on a new system.
- Proxy Edition– a Telnet proxy service is not created on an upgrade.
- MACH5 Edition–a transparent TCP tunnel connection listening on port 23 is created in place of the default Telnet proxy service.

To configure the Telnet Shell proxy to intercept traffic:

1. From the Management Console, select **Configuration > Services > Proxy Services**.

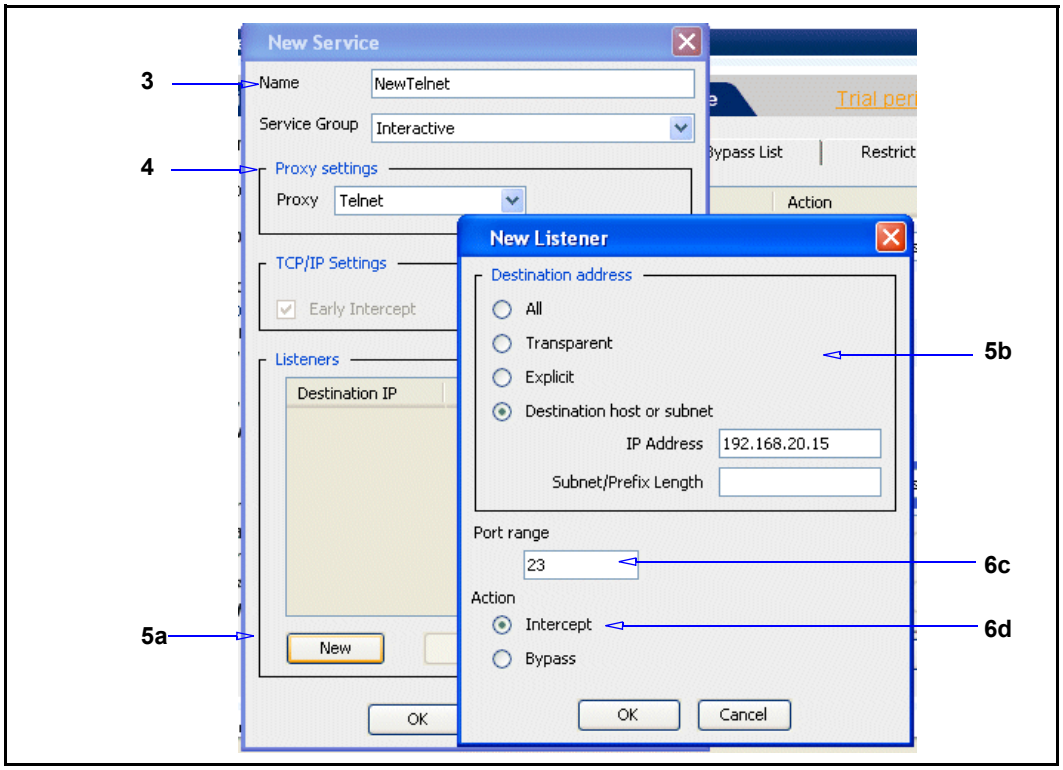


2. Scroll to the **Interactive** group and click it to expand the list. Click **Telnet**.
3. From the drop-down list, select **Intercept**.
4. Click **Apply**.

Creating or a New Telnet Shell Proxy Service

To create a Telnet proxy service:

1. Select **Configuration > Services > Proxy Services**.
2. To create a new proxy service, click **New Service**.



3. In the **Name** field, choose a meaning name for the new proxy service.
4. In the **Proxy settings** field, select **Telnet**.
5. Create a new listener:
   a. Click **New**.
   b. Select a Destination IP address from the radio buttons.
   c. In the **Port Range** field, enter the ports on which the service should listen. The default port is `23`.
   d. Select the default action for the service: **Bypass** tells the service to ignore any traffic. **Intercept** configures the service to intercept the traffic that is being proxied.
   e. Click **OK** to close the dialog.
6. Click **OK**.

Relevant CLI Syntax to Create/Edit a Telnet Proxy Service:

- To enter configuration mode:

```
SGOS#(config) proxy-services
SGOS#(config proxy-services) create telnet service-name
SGOS#(config proxy-services) edit service-name
```

- The following subcommands are available:

```
SGOS#(config service-name) add {transparent | explicit | all |
ip_address | ip_address/subnet-mask} {port | first_port-last_port}
[intercept | bypass]
```

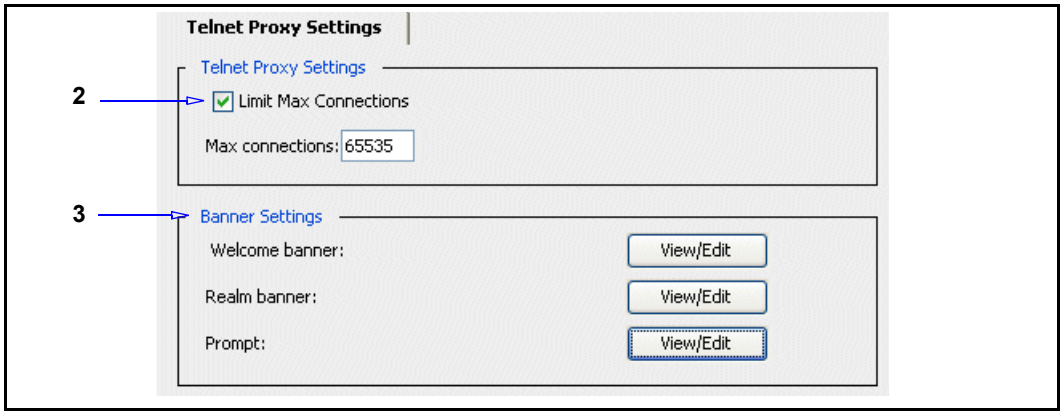
```
SGOS#(config service-name) bypass {transparent | explicit | all |
ip_address | ip_address/subnet-mask} {port | first_port-last_port}
SGOS#(config service-name) exit
SGOS#(config service-name) intercept {transparent | explicit | all |
ip_address | ip_address/subnet-mask} {port | first_port-last_port}
SGOS#(config service-name) remove {transparent | explicit | all |
ip_address | ip_address/subnet-mask} {port | first_port-last_port}
SGOS#(config service-name) view
```

Customizing Welcome and Realm Banners and Prompt Settings

You can configure banners for the Telnet shell and the realm and set the prompt that users see when entering the shell.

To customize Telnet shell proxy settings:

1. Select **Configuration > Proxy Settings > Shell Proxies > Telnet Proxy Settings.**



2. To set the maximum concurrent connections, select **Limit Max Connections**. Enter the number of maximum concurrent connections allowed for this service. Allowed values are between 1 and 65535.
3. (Optional) Change the default banner settings.
   - **Welcome banner**—Users see this when they enter the shell. The default string is: `Blue Coat $(module_name) proxy`.
   - **Realm banner**—Users see this help message just before they see the **Username** prompt for proxy authentication. The default string is: `Enter credentials for realm $(realm).`
   - **Prompt**—The command prompt. The default string is: `$(module_name)-proxy>`.

     For a list of available substitutions, see Table 10–1, "CPL Substitutions for Shell Proxies" on page 159.

     Click **View/Edit** to display the respective banner dialog. Change the string. Click **OK**.
4. Click **Apply.**

Related CPL Syntax to Customize Telnet Shell Proxy Settings

You can use CPL substitutions when creating welcome and realm banners and Telnet prompts. For a list of available CPL substitutions, see Table 10–1, "CPL Substitutions for Shell Proxies" on page 159.

Related CLI Syntax to Configure a Telnet Shell Proxy

```
SGOS#(config) shell {max-connections number_of_connections | prompt
prompt | realm-banner realm_banner | welcome-banner welcome_banner}
```

Notes for Telnet Shell Proxies

- Telnet credential exchange is in plaintext.
- A Telnet proxy cannot be used to communicate with non-Telnet servers (such as Webservers on port 80) because Telnet proxies negotiate Telnet options with the client before a server connection can be established.

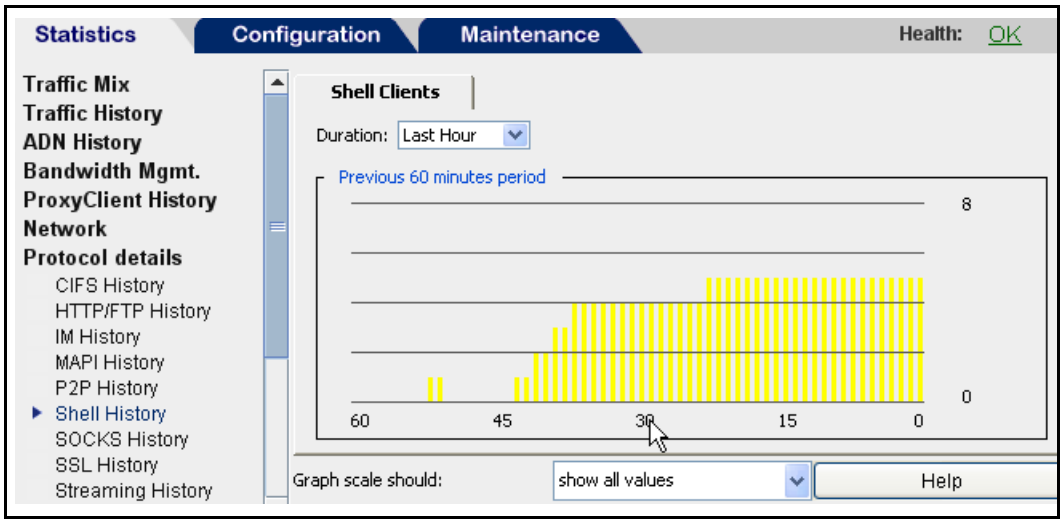
Viewing Shell History Statistics

The **Shell History** tab displays client connections over the last 60-minute, 24-hour, and 30-day period.

Note: The Shell history statistics are available only through the Management Console.

To view Shell history statistics:

1. Select **Statistics > Protocol Details > Shell History.**



2. Select a time- period for the graph from the **Duration:** drop-down list. The default setting is last hour.
3. (Optional) To set the graph scale to a different value, select a value from the **Graph scale should** drop-down list.

Chapter 11: Managing a SOCKS Proxy

This chapter discusses the ProxySG SOCKS proxy. While SOCKS servers are generally used to provide firewall protection to an enterprise, they also can be used to provide a generic way to proxy any TCP/IP or UDP protocols. The ProxySG supports both SOCKSv4/4a and SOCKSv5; however, because of increased username and password authentication capabilities and compression support, Blue Coat recommends that you use SOCKS v5.

Note: For Blue Coat compatibility with SOCKS clients, check with customer support. For information on the Permeo Premium Agent (Permeo PA), see "Using the Permeo PA SOCKS Client with the Blue Coat SOCKS Server" on page 172.

Topics in this Chapter

This chapter includes information about the following topics:

- "Creating or Editing a SOCKS Proxy Service" on page 168
- "Configuring the SOCKS Proxy" on page 170
- "Using Policy to Control the SOCKS Proxy" on page 171
- "Using the Permeo PA SOCKS Client with the Blue Coat SOCKS Server" on page 172
- "Viewing SOCKS History Statistics" on page 174

In a typical deployment, the SOCKS proxy works with the Endpoint Mapper proxy and MAPI handoff. In this deployment, you will:

- Create an Endpoint Mapper proxy at the remote office (the downstream proxy) that intercepts Microsoft RPC traffic and creates dynamic TCP tunnels. Traffic to port 135 is transparently redirected to this service using bridging or L4 switch or WCCP. For information on creating and enabling an Endpoint Mapper proxy service, see Chapter 6: "Accelerating the Microsoft Outlook Application (Endpoint Mapper and MAPI Proxies)" on page 83.
- Create any other TCP tunnel proxies you need at the remote office: SMTP, DNS, and the like. For information on configuring TCP tunnels, see Chapter 13: "Managing the TCP Tunneling Proxy" on page 207.
- Create a SOCKS gateway at the remote office and enable compression for that gateway. This SOCKS gateway points to a SOCKS proxy located at the main office location (the upstream proxy, the core of the network). For information on creating a SOCKS gateway and enabling SOCKS compression, see the SOCKS Gateway Configuration chapter in *Volume 5: Advanced Networking*.

- Set policy to forward TCP traffic through that SOCKS gateway. You can do this through the `<proxy>` layer using either the VPM or CPL. For more information, see "Using Policy to Control the SOCKS Proxy" on page 171.

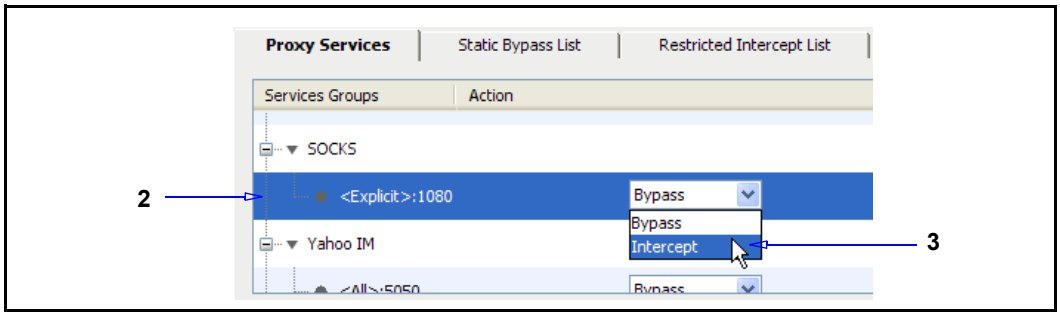
Configuring the SOCKS Proxy Service Options

This section describes how to change the default service options and add new services.

Changing the SOCKS Proxy Service to Intercept All IP Addresses on Port 1080

The service is configured to listen to all IP addresses, but is set in Bypass mode. The following procedure describes how to change the service to Intercept mode. To configure the SOCKS proxy to intercept traffic:

1. From the Management Console, select **Configuration > Services > Proxy Services**.
2. To edit an existing SOCKS proxy service, highlight the service and click **Edit**. To create a new proxy service, click **New**.

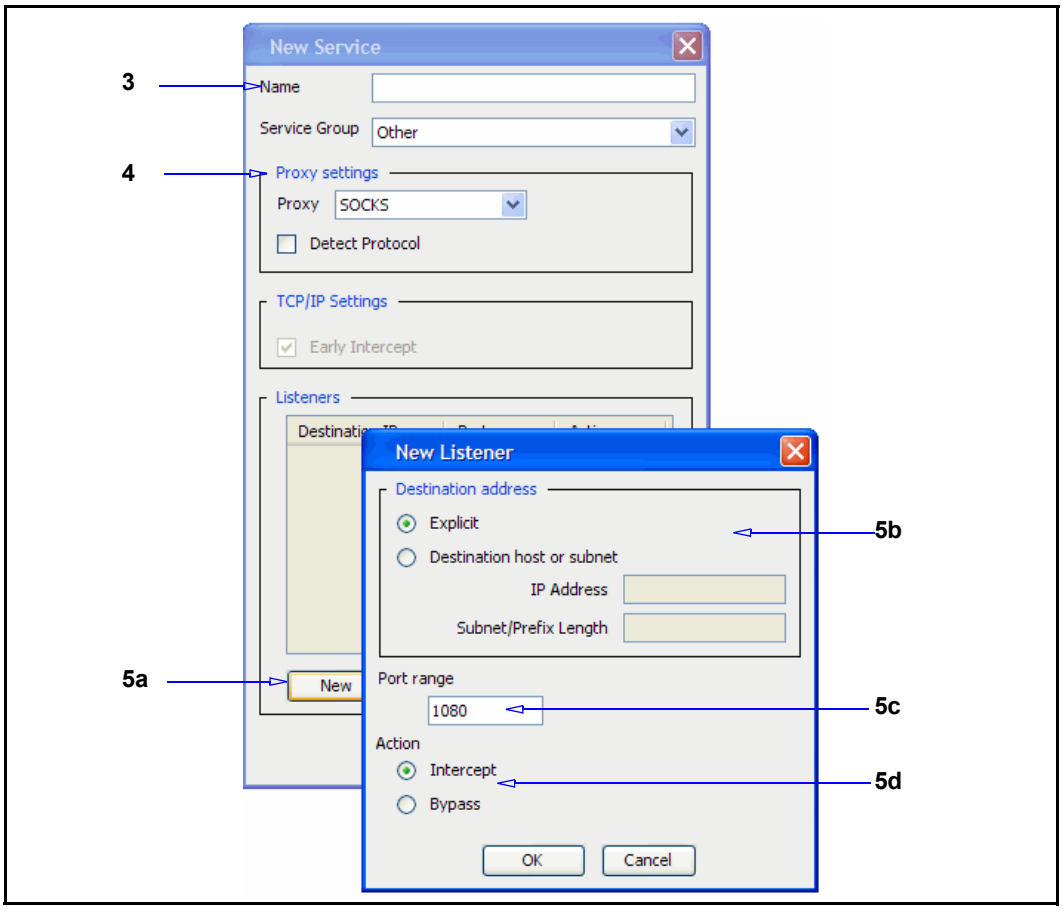


3. Scroll the list of services to display the default SOCKS service line; click the **+** symbol to expand the Shell services list.
4. Notice the **Action** for each default service (port 1080) is **Bypass**. Select **Intercept** from the drop-down list(s).
5. Click **Apply**.

Creating or Editing a SOCKS Proxy Service

To create or edit a SOCKS proxy service:

1. Select **Configuration > Services > Proxy Services**.
2. To edit an existing SOCKS proxy service, highlight the service and click **Edit**. To create a new proxy service, click **New**.



3. If you are creating a new SOCKS proxy service, enter a meaningful name in the **Name** field.
4. Configure Proxy Settings options:
   a. In the **Proxy settings** area, select SOCKS from the drop-down menu.
   b. Select the **Detect Protocol** checkbox to automatically detect the protocol being used. This breaks connections that do not have the client send information first, but expect the server to respond on connection. It also can add significant delay if the client does not send specific information, and only after timing out does it treat the traffic as unknown.
5. Create a new listener:
   a. Click **New**; if you edit an existing listener, click **Edit**.
   b. Define the IP address option: explicit or the specified address.
   c. In the **Port Range** field, enter the ports on which the service listens. The default port for the SOCKS proxy is 1080.
   d. Select the default behavior for the service: **Bypass** tells the service to ignore any traffic. **Intercept** configures the service to intercept the traffic that is being proxied.

e. Click **OK** to close the New Listener dialog.

6. Click **OK** to close the New Service dialog.
7. Click **Apply.**

Relevant CLI Syntax to Create/Edit a Proxy Service:

- To enter configuration mode for the service:

```
SGOS#(config) proxy-services
SGOS#(config proxy-services) create socks service-name
SGOS#(config proxy-services) edit service-name
```

- The following subcommands are available:

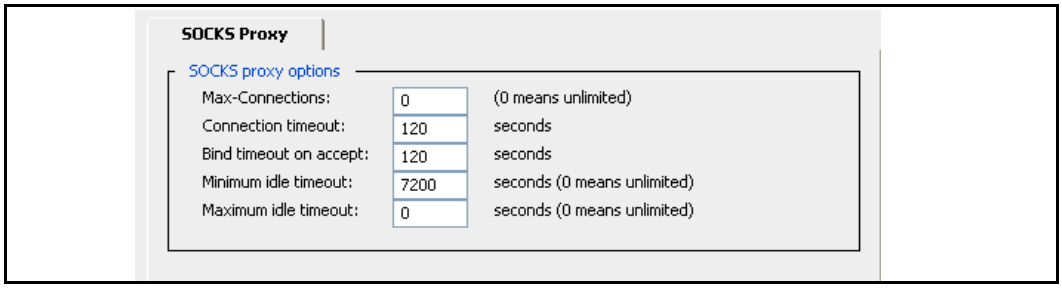
```
SGOS#(config service-name) add {explicit | ip_address | ip_address/
subnet-mask} {port | first_port-last_port} [intercept | bypass]
SGOS#(config service-name) attribute {adn-optimize {enable | disable}|
detect-protocol {enable | disable}}
SGOS#(config service-name) bypass {explicit | ip_address | ip_address/
subnet-mask} {port | first_port-last_port}
SGOS#(config service-name) intercept {explicit | ip_address |
ip_address/subnet-mask} {port | first_port-last_port}
SGOS#(config service-name) remove {explicit | ip_address | ip_address/
subnet-mask} {port | first_port-last_port}
SGOS#(config service-name) view
```

Configuring the SOCKS Proxy

Complete the following steps to create a SOCKS proxy and to configure SOCKS-proxy connection and timeout values.

To create a SOCKS proxy server:

1. Select **Configuration > Services > SOCKS Proxy.**



2. Fill in the option fields (described below) as needed. The defaults are displayed and should be sufficient for most purposes.

Table 11–1 SOCKS Proxy Options

Option	Suboption	Description
Max-Connections	*connections*	Set maximum allowed SOCKS client connections. The default of 0 indicates an infinite number of connections are allowed.
Connection timeout	*seconds*	Set maximum time to wait on an outbound CONNECT.
Bind timeout on accept	*seconds*	Set maximum time to wait on an inbound BIND.
Minimum idle timeout	*seconds*	Specifies the minimum timeout after which SOCKS can consider the connection for termination when the max connections are reached.
Maximum idle timeout	*seconds*	Specifies the max idle timeout value after which SOCKS should terminate the connection.

Related CLI Syntax to Configure the SOCKS Proxy

```
SGOS#(config) socks-proxy accept-timeout seconds
SGOS#(config) socks-proxy connect-timeout seconds
SGOS#(config) socks-proxy max-connections num_connections
SGOS#(config) socks-proxy max-idle-timeout seconds
SGOS#(config) socks-proxy min-idle-timeout seconds
```

Using Policy to Control the SOCKS Proxy

Once the basic configuration for the SOCKS proxy has been set, you can use policy to control the SOCKS proxy.

- To use SOCKS version 5, which allows you to use a SOCKS username/password, you must set the version through policy.
  - If using VPM, go to the Forwarding layer, select **Source > Set Source Object > New > SOCKS Version.**
  - If using CPL, enter the following:

```
<proxy> client.protocol=socks
  ALLOW socks.version=5
  DENY
```

- If browsers and FTP clients are configured to use SOCKS encapsulation and a rule in policy is matched that denies a transaction, you will see a **page cannot be displayed** message instead of an exception page.

  This is expected behavior, as a deny action abruptly closes the client's TCP connection, yet the client is expecting a SOCKS-style closure of the connection. You can avoid this, and return an exception page by applying the following policy:

- If using VPM, go to the Web Access layer, create two rules. For the first rule, select **Service > New > Client Protocol > SOCKS > TCP Tunneling over SOCKS**; for the second, select **Service > New > Client Protocol > SOCKS > All SOCKS.**
- If using CPL, enter the following:

```
<Proxy>
  DENY socks=yes tunneled=yes
  DENY socks=yes
```

Using the Permeo PA SOCKS Client with the Blue Coat SOCKS Server

The ProxySG can be used as a SOCKS gateway by the Permeo Premium Agent (PA), with full licensing support and Dynamic Port Management (DPM) functionality.

The ProxySG supports the Windows Permeo PA SOCKS client version 5.12a, including those clients that require the special probe license protocol and corresponding customer ID. Note that each ProxySG can only support PA clients with the same customer ID.

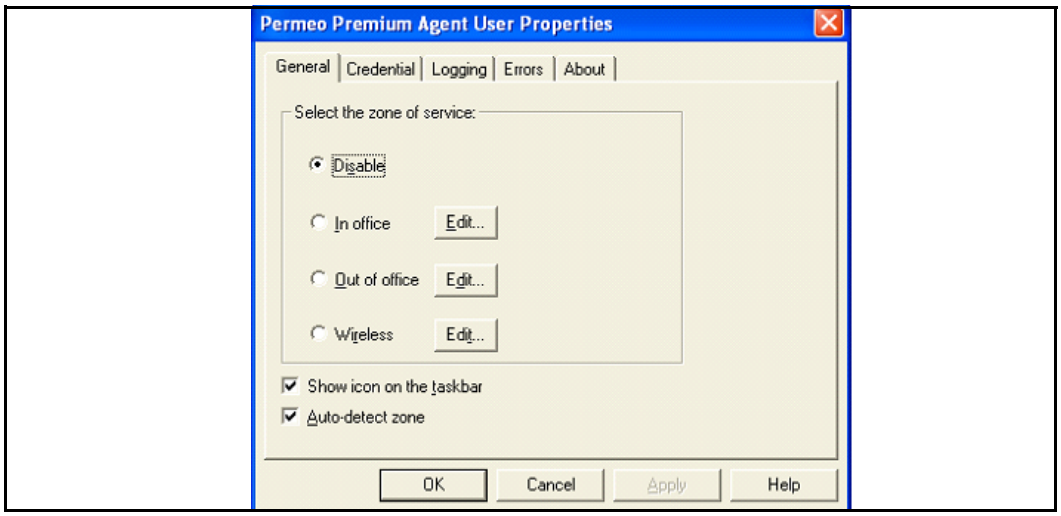
Licensing the PA SOCKS client on the ProxySG is a two-step process:

- Get the customer ID from the PA client.
- Tell the ProxySG the PA customer ID.

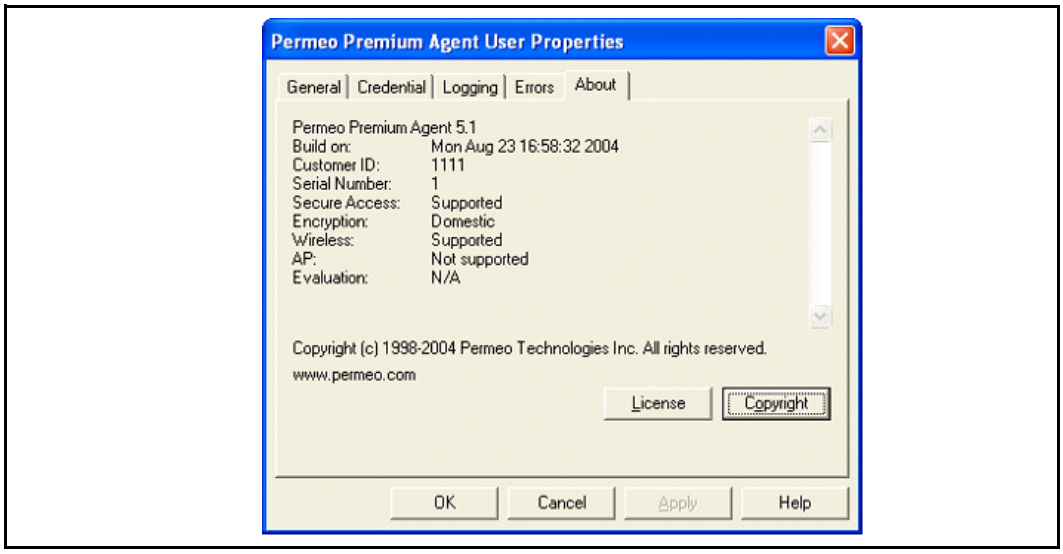
Note: The default license setting for the Permeo PA client on the ProxySG is off. This setting should only be enabled when you are using the PA client.

To obtain the PA Customer ID:

1. From the PA client, launch the Permeo Agent User Properties (**Start Menu > All Programs > Permeo Premium Agent**).



2. Click the **About** tab.



3. Make a note of the Customer ID number, which is in hex. In the example above the **Customer ID** is 1111.

To validate the Permeo PA license on the ProxySG:

Note: You cannot validate the license through the Management Console.

1. From the ProxySG, launch the CLI:

```
SGOS> enable
Enable Password:
SGOS# configure terminal
Enter configuration commands, one per line. End with CTRL-Z.
```

2. From the (`config`) prompt:

```
SGOS#(config) socks-proxy pa-customer-id customer_id
```

where *`customer_id`* is the Customer ID number you took from the **About** tab on the PA client.

To disable the Permeo PA license:

From the (`config`) prompt:

```
SGOS#(config) socks-proxy pa-customer-id 0
```

Limitations

- Protocol Detection interferes with SOCKS and must be disabled on the ProxySG. The CPL policy should include the line `detect_protocol(no)`.
- SOCKS compression should be disabled when using the PA SOCKS client. The CPL policy should include the line `socks.accelerate(no)`.
- The ProxySG only supports username and password authentication between the ProxySG and the SOCKS Permeo PA client.

- The ping and trace route functions from Permeo PA administrator tool are not compatible with this release (5.1).
- Proxy chaining is not supported between the ProxySG and the Permeo Application Gateway (ASG).
- The policy update feature on the PA is not supported when using the ProxySG. PA can get policy from the HTTP source as well as the ASG so it can still perform automatic updates from a external Web server.
- Only the UPWD authentication method is supported.

Viewing SOCKS History Statistics

The **SOCKS History** tabs (SOCKS Clients, SOCKS Connections, and SOCKS client and server compression) display client data, Connect, Bind, and UPD Associate requests, client and server UDP, TCP and compression requests.

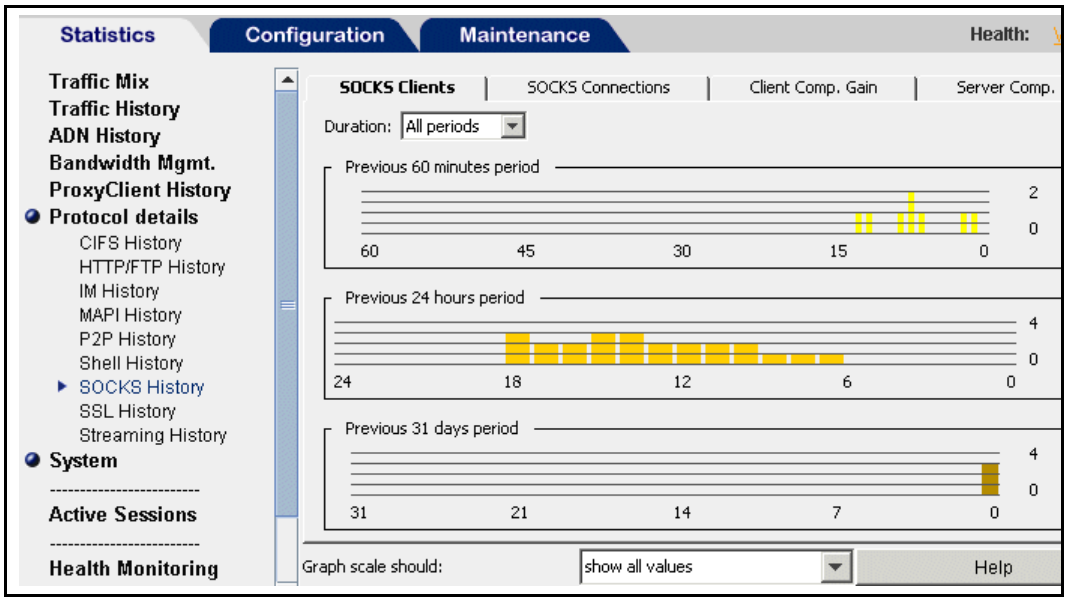
Note: The SOCKS history statistics are available only through the Management Console.

Viewing SOCKS Clients

The SOCKS Clients tab displays SOCKS Client data.

To view SOCKS client data:

1. Select **Statistics > SOCKS History > SOCKS Clients**.
2. Select a time period for the graph from the **Duration:** drop-down list.
3. (Optional) To set the graph scale to a different value, select a value from the **Graph scale should** drop-down list.

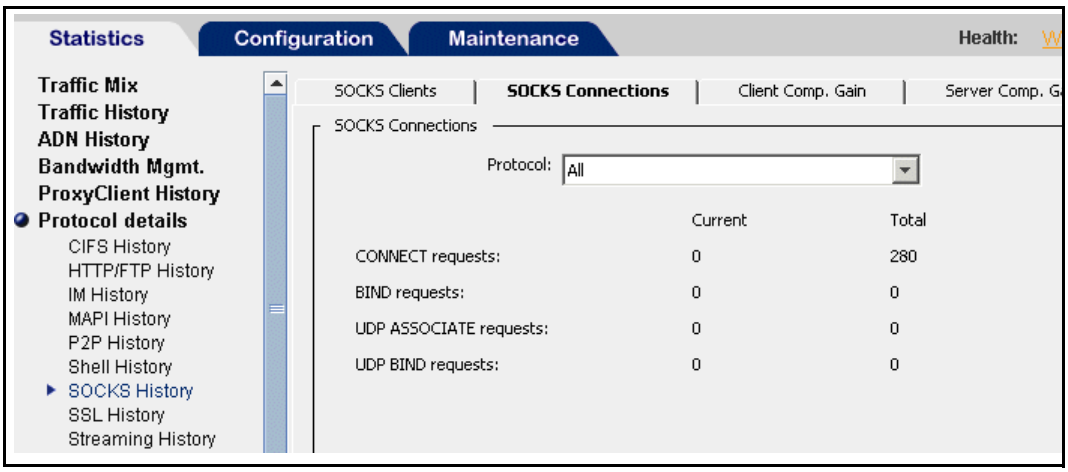


Viewing SOCKS Connections

The SOCKS Connections tab displays SOCKS Connection data.

To view SOCKS connection data:

Select **Statistics > SOCKS History > SOCKS Connections.**



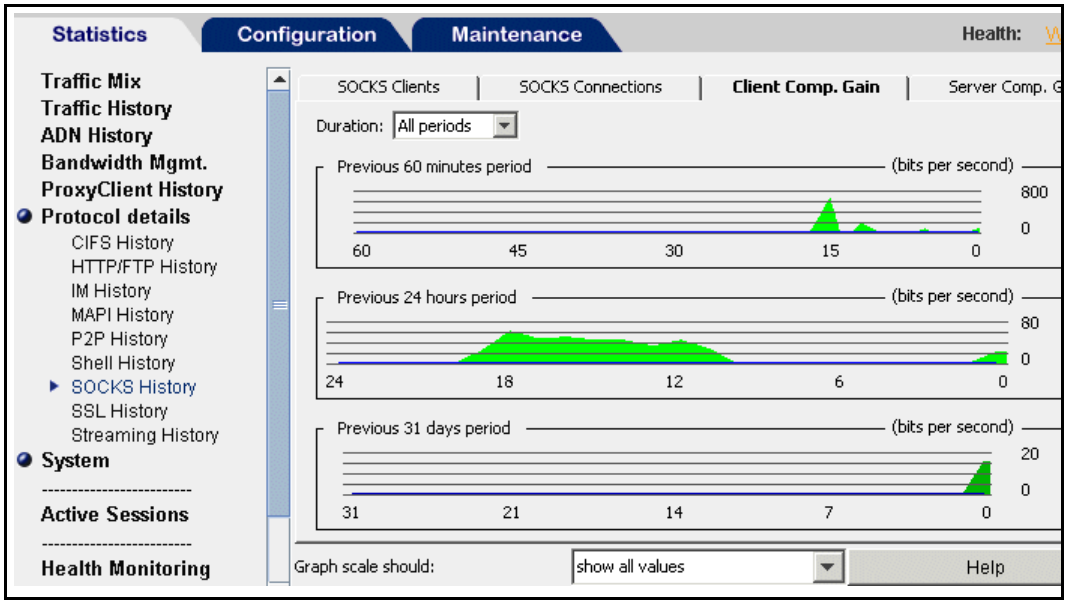
Viewing SOCKS Client and Server Compression Gain Statistics

You can view SOCKS client and server compression-gain statistics for the ProxySG over the last 60 minutes, 24 hours, and 30 days in the **Client Comp. Gain** and the **Server Comp. Gain** tabs. These statistics are not available through the CLI.

The green display on the bar graph represents uncompressed data; the blue display represents compressed data. Hover your cursor over the graph to see the compressed gain data.

To view SOCKS client compressed gain statistics:

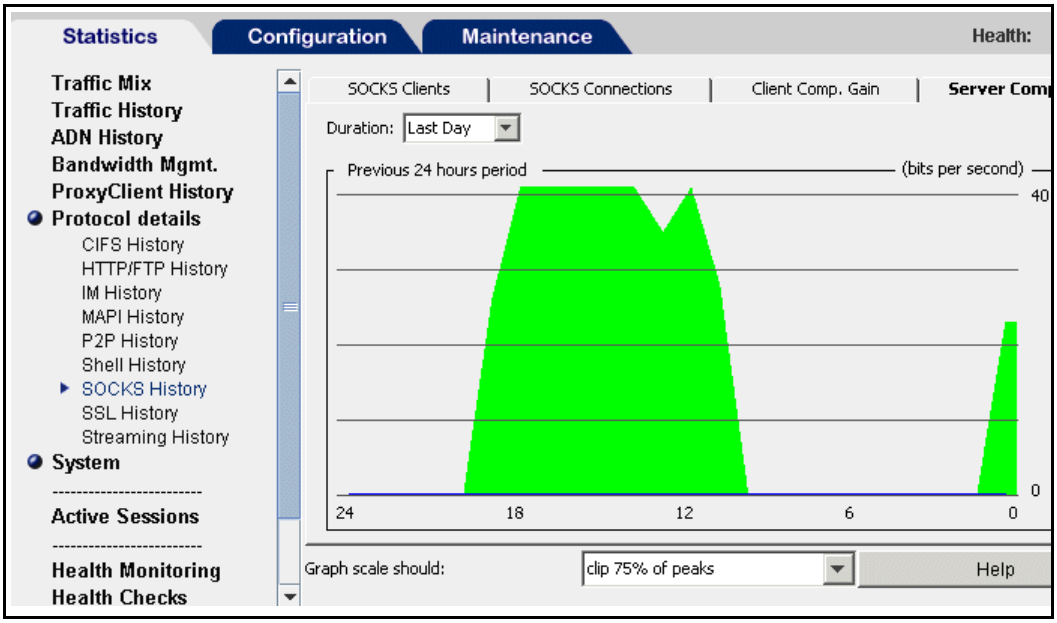
1. Select **Statistics > SOCKS History > Client Comp. Gain**.
2. Select a time priod for the graph from the **Duration:** drop-down list.



3. (Optional) To set the graph scale to a different value, select a value from the **Graph scale should** drop-down list.

To view SOCKS Server compressed gain statistics:

1. Select **Statistics > SOCKS History > Server Comp. Gain.**
2. Select a time period from the **Duration:** drop-down list.



3. (Optional) To set the graph scale to a different value, select a value from the **Graph scale should** drop-down list.

Chapter 12: Managing the SSL Proxy

This chapter discusses the ProxySG SSL proxy. HTTPS traffic poses a major security risk to enterprises. Because the SSL content is encrypted, it cannot be monitored by normal means, allowing users to bring in viruses, access forbidden sites, or leak confidential business information over the HTTPS connection on port 443.

The SSL proxy allows you to intercept HTTPS traffic (in explicit and transparent modes) so that security measures such as authentication, virus scanning and URL filtering, and performance enhancements such as HTTP caching can be applied to HTTPS content. Additionally, the SSL proxy allows you to validate server certificates presented by various HTTPS sites at the gateway and offers information about the HTTPS traffic in the access log.

Topics in this Chapter

This chapter includes information about the following topics:

- Section A: "Intercepting HTTPS Traffic" on page 181
- Section B: "Configuring SSL Rules through Policy" on page 190
- Section C: "Viewing SSL Statistics" on page 197
- Section D: "Advanced Topics" on page 200

For information on Certificate Authority (CA) certificates, keyrings, and keypairs, refer to *Volume 4: Securing the Blue Coat ProxySG Appliance*.

Understanding the SSL Proxy

The SSL proxy can be used to tunnel or intercept HTTPS traffic. The SSL proxy tunnels all HTTPS traffic by default unless there is an exception, such as a certificate error or a policy denial. In such cases the SSL proxy intercepts the SSL connection and sends an error page to the user. The SSL proxy allows interception of HTTPS traffic for monitoring reasons as well.

Note: Some HTTPS traffic, such as financial information, should not be intercepted.

The SSL proxy can do the following operations while tunneling HTTPS traffic.

- Validate server certificates, including revocation checks using Certificate Revocation Lists (CRLs).
- Check various SSL parameters such as cipher and version.
- Log useful information about the HTTPS connection.

When the SSL proxy is used to intercept HTTPS traffic, it can also:

- Cache HTTPS content.
- Apply HTTP-based authentication mechanism.
- Do virus scanning and URL filtering.
- Apply granular policy (such as validating mime type and filename extension).

Validating the Server Certificate

The SSL proxy can do the following checks on server certificates:

- Verification of issuer signature.
- Verification of certificate dates.
- Comparison of hostname in the URL and certificate (intercepted connections only).

  Hostnames in server certificates are important because the SSL proxy can identify a Web site just by looking at the server certificate if the hostname is in the certificate. Most content-filtering HTTPS sites follow the guideline of putting the name of the site as the common name in the server's certificate.
- Verification of revocation status.

  To mimic the overrides supported by browsers, the SSL proxy can be configured to ignore failures for the verification of issuer signatures and certificate dates and comparison of the hostname in the URL and the certificate.

The ProxySG trusts all root CA certificates that are trusted by Internet Explorer and Firefox. This list is updated to be in sync with the latest versions of IE and Firefox.

Checking CRLs

An additional check on the server certificate is done through Certificate Revocations Lists (CRLs). CRLs show which certificates are no longer valid; the CRLs are created and maintained by Certificate Signing Authorities that issued the original certificates.

Only CRLs that are issued by a trusted issuer can be used by the ProxySG. The CRL issuer certificate must exist as CA certificate on the ProxySG before the CRL can be imported.

The ProxySG allows:

- One local CRL per certificate issuing authority.
- An import of a CRL that is expired; a warning is displayed in the log.
- An import of a CRL that is effective in the future; a warning is displayed in the log.

Determining What HTTPS Traffic to Intercept

The SSL proxy tunnels HTTPS traffic by default; it does not intercept HTTPS traffic.

Many existing policy conditions, such as destination IP address and port number can be used to decide which HTTPS connections to intercept.

Additionally, the SSL proxy allows the hostname in the server certificate to be used to make the decision to intercept or tunnel the traffic. The server certificate hostname can be used as is to make intercept decisions for individual sites, or it can be categorized using any of the various URL databases supported by Blue Coat.

Categorization of server certificate hostnames can help place the intercept decision for various sites into a single policy rule.

Recommendations for intercepting traffic include:

- Intercept Intranet traffic.
- Intercept suspicious Internet sites, particularly those that are categorized as none in the server certificate.

Managing Decrypted Traffic

After the HTTPS connection is intercepted, you can do:

- Anti-virus scanning over ICAP.
- URL filtering (on box and off-box). Blue Coat recommends on box URL / content filtering if you use transparent proxy. When the URL is sent off-box for filtering, only the hostname or IP address of the URL (not the full path) is sent for security reasons.
- Filtering based on the server certificate hostname.
- Caching.

HTTPS applications that require browsers to present client certificates to secure Web servers do not work if you are intercepting traffic. Create a policy rule to prevent the interception of such applications.

If you intercept HTTPS traffic, be aware that local privacy laws might require you to notify the user about interception or obtain consent prior to interception. You can use the HTML Notify User object to notify users after interception. You can use consent certificates to obtain consent prior to interception. The HTML Notify User is easier; however, the ProxySG must decrypt the first request from the user before it can issue an HTML notification page.

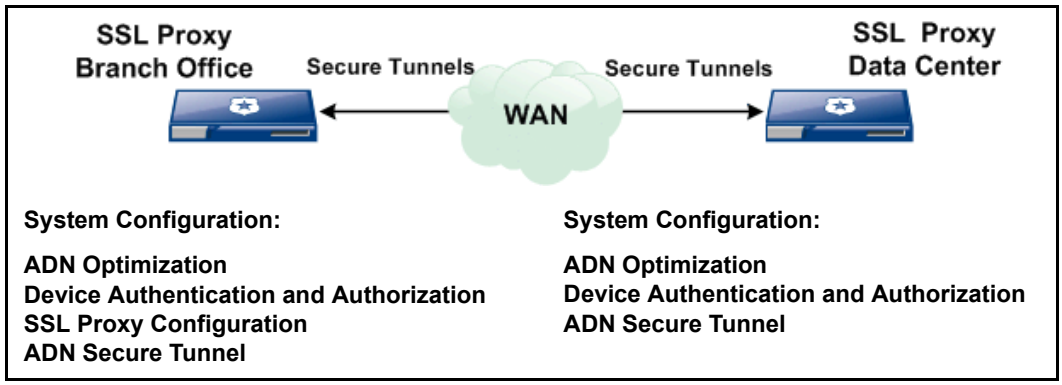
Using the SSL Proxy with ADN Optimization

The SSL proxy itself can be used as a split proxy, which requires two SSL proxies, one at the branch and one at the core, working together. A *split proxy* can be configured (see below) to implement functionality that is not possible in a standalone proxy.

In this configuration, the SSL proxy supports ADN optimization on WAN networks, and SSL traffic performance can be increased through the byte caching capability offered. The branch proxy, which makes the decisions, is configured with both ADN optimization and SSL proxy functionality.

The concentrator proxy (a ProxySG that provides access to data center resources) does not require any configuration related to the SSL proxy. It only requires the necessary ADN configuration for applying byte caching capabilities to intercepted SSL content.

No special configuration is required to the SSL proxy. Securing the tunnels and authenticating the devices occurs from the **Configuration > ADN** panes.



Section A: Intercepting HTTPS Traffic

Intercepting HTTPS traffic (by decrypting SSL connections at the ProxySG) allows you to apply security measures like virus scanning and URL filtering.

Configuration to intercept HTTPS traffic requires the following tasks:

- Determine whether you are using transparent or explicit mode. For information on explicit versus transparent proxies, see Appendix A: "Explicit and Transparent Proxy" on page 213.
- Create an SSL service or HTTP / SOCKS services with protocol detection enabled, depending on whether you are using transparent or explicit mode. For more information on creating an SSL service, skip to "Setting Up the SSL Proxy in Transparent Proxy Mode" on page 182.
- Create or import an issuer keyring, which is used to sign emulated server certificates to clients on the fly, allowing the SSL proxy to examine SSL content. For more information on creating an issuer keyring, see "Specifying an Issuer Keyring and CCL Lists for SSL Interception" on page 185.
- (Optional) Use the **Notify User** object or client consent certificates to notify users that their requests are being intercepted and monitored. Whether this is required depends on local privacy laws. Note that the ProxySG has to decrypt the first request from the user to issue an HTML notification page. If this is not desirable, use client consent certificates instead. For more information on configuring the **Notify User** object, refer to *Volume 7: Managing Content*. For information on managing client consent certificates, see "Using Client Consent Certificates" on page 186.
- Download CA certificates to desktops to avoid a security warning from the client browsers when the ProxySG is intercepting HTTPS traffic. For information, see "Downloading an Issuer Certificate" on page 186.
- Using policy (VPM or CPL), create rules to intercept SSL traffic and to control validation of server certificates. By default, such traffic is tunneled and not intercepted. You must create suitable policy before intercepting SSL traffic. For more information on using policy to intercept SSL traffic, see Section B: "Configuring SSL Rules through Policy" on page 190.
- Configure the Blue Coat AV or other third-party ICAP vendor, if you have not already done this. For more information on ICAP-based virus scanning, refer to *Volume 7: Managing Content*.
- Configure the Blue Coat Web Filter (BCWF) or a third-party URL-filtering vendor, if you have not already done this. For more information on configuring BCWF, refer to *Volume 7: Managing Content*.
- Configure Access Logging. For more information on configuring access logging, refer to *Volume 8: Access Logging*.
- Customize Exception Pages: To customize exception pages (in case of server certificate verification failure), refer to *Volume 6: The Visual Policy Manager and Advanced Policy*.

Section A: Intercepting HTTPS Traffic

Setting Up the SSL Proxy in Transparent Proxy Mode

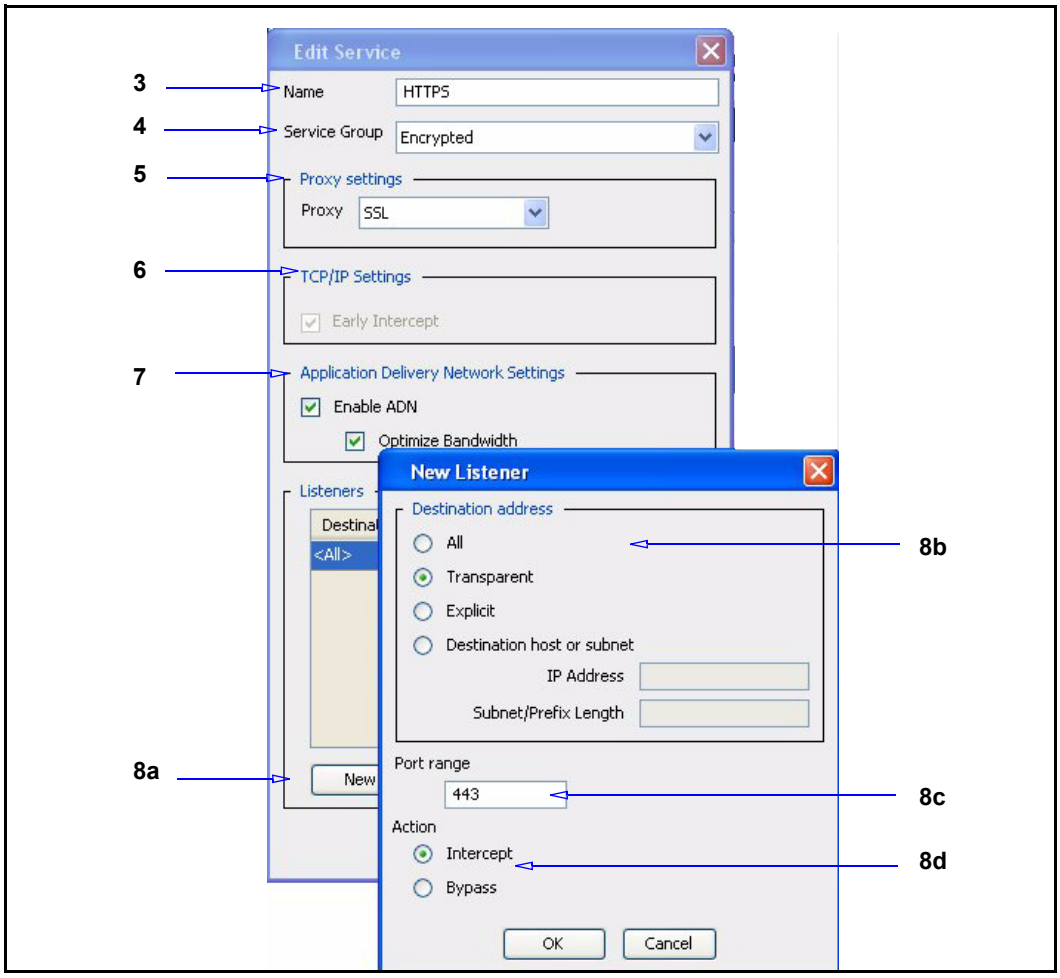
Proxy services are configured from the Management Console or the CLI. If using the SSL proxy in transparent mode, continue with this section.

If you are using the SSL proxy in explicit mode, you might need an HTTP proxy or a SOCKS proxy. For information on configuring an SSL proxy in explicit mode, see "Setting Up the SSL Proxy in Explicit Proxy Mode" on page 184.

You can use a TCP Tunnel service in transparent mode to get the same functionality. A TCP tunnel service is useful when you have a combination of SSL and non-SSL traffic going over port 443 and you do not want to break the non-SSL traffic. The SSL service requires that all requests to its port be SSL.

To configure an SSL service in transparent proxy mode:

1. From the Management Console, select **Configuration > Services > Proxy Services.**
2. Click **New.**



3. Give the SSL proxy a meaningful name.
4. Select the **Service Group** from the drop-down list. By defalt, **Other** is selected.
5. Select **SSL** from the **Proxy settings** drop-down list.
6. Configure TCP / IP Settings option:The **Early Intercept** option cannot be changed for the SSL proxy service.
7. Select ADN options:
   - **Enable ADN**. Select this option if you want this service to use ADN. Enabling ADN does not guarantee the connections are accelerated by ADN. The actual enable decision is determined by ADN routing (for explicit deployment) and network setup (for transparent deployment).

- The **Optimize Bandwidth** option is selected by default if you enabled WAN optimization during initial configuration. Clear the option if you are not configuring WAN optimization.

8. Create a new listener:
   a. Click **New**; if you edit an existing listener, click **Edit**.
   b. Define the IP address option: explicit or the specified address.
   c. In the **Port Range** field, enter the ports on which the service should listen. The default port for SSL is `443`.
   d. Select the default behavior for the service: **Bypass** tells the service to ignore any traffic. **Intercept** configures the service to intercept the traffic that is being proxied.
   e. Click **OK** to close the dialog.
9. Click **OK** to close the Edit Service dialog.
10. Click **Apply**.

Continue with "Specifying an Issuer Keyring and CCL Lists for SSL Interception" on page 185.

Related CLI Syntax to Create/Edit an SSL Proxy Service:

- To enter configuration mode for the service:

```
SGOS#(config) proxy-services
SGOS#(config proxy-services) create service-type service-name
SGOS#(config proxy-services) edit service-name
```

- The following subcommands are available:

```
SGOS#(config service-name) add {transparent | ip_address | ip_address/
subnet-mask} {port | first_port-last_port} [intercept | bypass]
SGOS#(config service-name) attribute {adn-optimize {enable | disable}
| use-adn {enable | disable}}
SGOS#(config service-name) bypass {transparent | ip_address |
ip_address/subnet-mask} {port | first_port-last_port}
SGOS#(config service-name) exit
SGOS#(config service-name) intercept {transparent | ip_address |
ip_address/subnet-mask} {port | first_port-last_port}
SGOS#(config service-name) remove {transparent | ip_address |
ip_address/subnet-mask} {port | first_port-last_port}
SGOS#(config service-name) view
```

Setting Up the SSL Proxy in Explicit Proxy Mode

The SSL proxy can be used in explicit mode in conjunction with the HTTP Proxy or SOCKS Proxy. You must create an HTTP Proxy service or a SOCKS Proxy service and use it as the explicit proxy from desktop browsers. You must also ensure that the detect-protocol attribute is enabled for these services.

When requests for HTTPS content are sent to either a SOCKS proxy or an HTTP proxy, the proxies can detect the use of the SSL protocol on such connections and enable SSL proxy functionality.

For information on configuring a new explicit HTTP or SOCKS proxy service, see "Creating an Explicit Proxy Server" on page 214.

Continue with "Specifying an Issuer Keyring and CCL Lists for SSL Interception" on page 185.

Specifying an Issuer Keyring and CCL Lists for SSL Interception

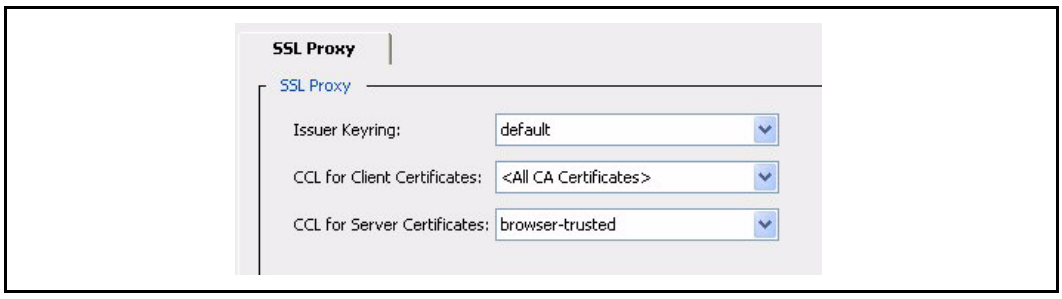
The SSL proxy can emulate server certificates; that is, present a certificate that appears to come from the origin content server. In actuality, Blue Coat has emulated the certificate and signed it using the issuer keyring. By default only the `subjectName` and the expiration date from the server certificate are copied to the new certificate sent to the client.

Note: Only keyrings with both a certificate and a keypair can be used as issuer keyrings.

You can also change the CA Certificate Lists (CCLs) that contain the CAs to be trusted during client and server certificate validation. The defaults are adequate for the majority of situations. For more information about CCLs, refer to *Volume 4: Securing the Blue Coat ProxySG Appliance*.

To specify the keyring and CCLs:

1. From the Management Console, select **Configuration > Proxy Settings > SSL Proxy.**



2. **Issuer Keyring:** From the drop-down menu, select the keyring to use as the issuer keyring. Any keyring with both a certificate and a keypair in the drop-down menu can be used.
3. **CCL for Client Certificates:** Choose which CAs are trusted when the SSL proxy validates client certificates. The default is **<All CA Certificates>**.
4. **CCL for Server Certificates:** Choose which CAs are trusted when the SSL proxy validates server certificates.The CCL for server certificates is relevant even when SSL proxy is tunneling SSL traffic. The default is **browser-trusted**.
5. Click **Apply.**

To configure policy, see "Configuring SSL Rules through Policy" on page 190.

Related CLI Syntax to Specify the Keyring and CCL Lists

This procedure assumes a keyring has already been created.

```
SGOS#(config ssl) proxy issuer-keyring keyring_name
SGOS#(config ssl) proxy client-cert-ccl {ccl_list_name | all | none}
SGOS#(config ssl) proxy server-cert-ccl {ccl_list_name | all}
```

Using Client Consent Certificates

The SSL proxy, in forward proxy deployments, can specify whether a client (typically a browser) certificate is required. These certificates are used for user consent, not for user authentication. Whether they are needed depends upon local privacy laws.

With client consent certificates, each user is issued a pair of certificates with the corresponding private keys. Both certificates have a meaningful user-readable string in the **common name** field. One certificate has a string that indicates grant of consent something like: “Yes, I agree to SSL interception”. The other certificate has a common name indicating denial of consent, something like: “No, I do not agree to SSL interception”.

Policy is installed on the ProxySG to look for these common names and to allow or deny actions. For example, when the string “Yes, I agree to SSL interception” is seen in the client certificate common name, the connection is allowed; otherwise, it is denied.

To configure client consent certificates:

1. Install the issuer of the client consent certificates as a CA certificate.
2. In VPM, configure the **Require Client Certificate** object in the **SSL Layer > Action** column.
3. Configure the **Client Certificate** object in the **Source** column to match common names.

Downloading an Issuer Certificate

When the SSL proxy intercepts an SSL connection, it presents an emulated server certificate to the client browser. The client browser issues a security pop-up to the end-user because the browser does not trust the issuer used by the ProxySG. This pop-up does not occur if the issuer certificate used by SSL proxy is imported as a trusted root in the client browser's certificate store.

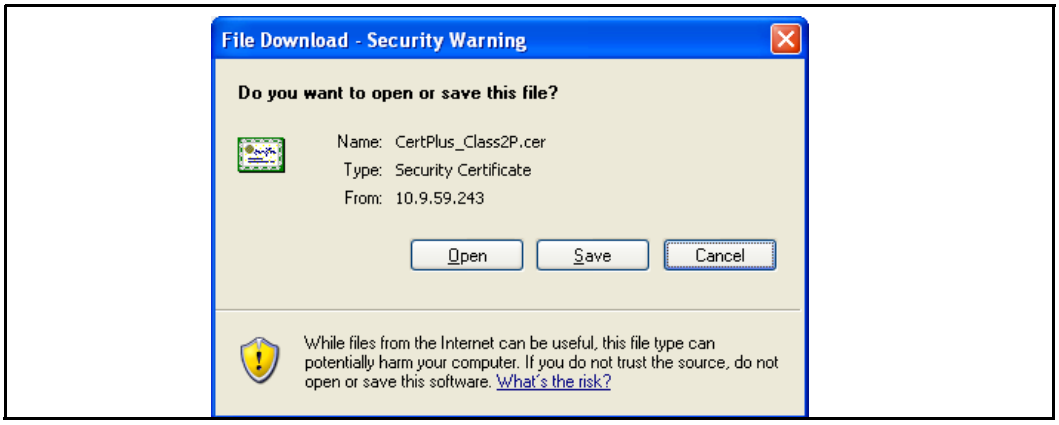
The ProxySG makes all configured certificates available for download via its management console. You can ask end users to download the issuer certificate through Internet Explorer or Firefox and install it as a trusted CA in their browser of choice. This eliminates the certificate popup for emulated certificates.

To download the certificate through Internet Explorer, see "To download a certificate through Internet Explorer:" on page 187. To download a certificate through Firefox, see "To download a certificate through Firefox:" on page 188.

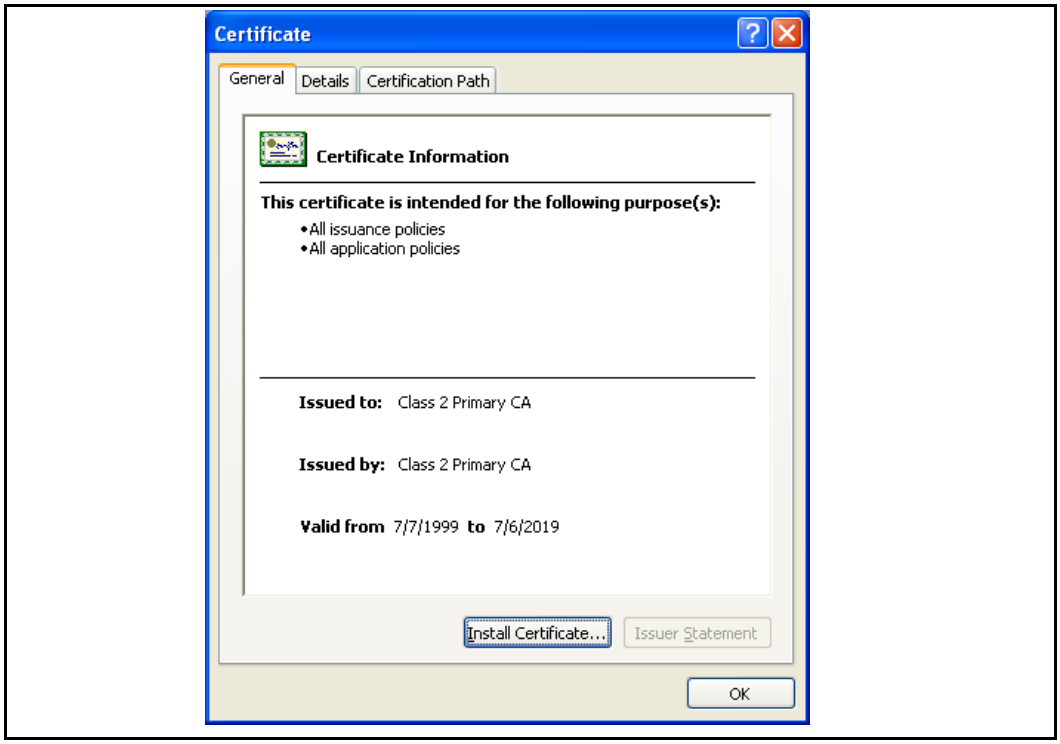
To download a certificate through Internet Explorer:

Note: You can e-mail the console URL corresponding to the issuer certificate to end users so that the he or she can install the issuer certificate as a trusted CA.

1. Select **Statistics > Advanced**.
2. Select **SSL**.
3. Click **Download a Certificate as a CA Certificate**; the list of certificates on the system display.
4. Click a certificate (it need not be associated with a keyring); the File Download Security Warning displays asking what you want to do with the file.



5. Click **Save**. When the **Save As** dialog box displays, click **Save**; the file downloads.
6. Click **Open** to view the Certificate properties; the Certificate window displays.



7. Click the **Install Certificate** button to launch the **Certificate Import Wizard**.
8. Ensure the **Automatically select the certificate store based on the type of certificate** radio button is enabled before completing the wizard
9. Click **Finish**. the wizard announces when the certificate is imported.
10. (Optional) To view the installed certificate, go to Internet Explorer, **Select Tools > Internet Options > Contents > Certificates**, and open either the **Intermediate Certification Authorities** tab or the **Trusted Root Certification Authorities** tab, depending on the certificate you downloaded.

To download a certificate through Firefox:

Note: You can e-mail the console URL corresponding to the issuer certificate to end users so that the end-user can install the issuer certificate as a trusted CA.

---

1. Select **Statistics > Advanced**.
2. Select **SSL**.
3. Click **Download a ProxySG Certificate as a CA Certificate**; the list of certificates on the system display.
4. Click a certificate (it need not be associated with a keyring); the **Download Certificate** dialog displays.

5. Enable the options needed. View the certificate before trusting it for any purpose.
6. Click **OK**; close the Advanced Statistics dialog.

Section B: Configuring SSL Rules through Policy

SSL interception and access rules, including server certificate validation, are configured through policy—either the VPM or CPL. Use the **SSL Intercept Layer** to configure SSL interception; use the **SSL Access Layer** to control other aspects of SSL communication such as server certificate validation and SSL versions. To configure SSL rules using CPL, skip to "CPL in the SSL Intercept Layer" on page 194.

This section covers the following topics:

- "Using the SSL Intercept Layer" on page 190.
- "Using the SSL Access Layer" on page 192
- "Using Client Consent Certificates" on page 186

Using the SSL Intercept Layer

The SSL intercept layer allows you to set intercept options:

- "To intercept HTTPS content through VPM:" on page 190
- "To intercept HTTPS requests to specific sites through VPM:" on page 191
- "To customize server certificate validation through VPM:" on page 193

For a list of policy conditions, properties, and actions, see "CPL in the SSL Intercept Layer" on page 194.

Note: For detailed instructions on using VPM, refer to *Volume 6: The Visual Policy Manager and Advanced Policy*.

To intercept HTTPS content through VPM:

1. Select **Configuration > Policy > Visual Policy Manager** and launch VPM.
2. From the **Policy** drop-down menu, select **Add SSL Intercept** Layer.
3. Right-click **Set** in the **Action** column; the **Set Action** object displays.
4. Click **New** and select **Enable HTTPS Intercept** object or the **Enable HTTPS Intercept on Exception** object.

The checkboxes for **Issuer Keyring**, **Hostname**, **Splash Text**, and **Splash URL** all control various aspects for certificate emulation. Fill in the fields as follows:

a. **Issuer Keyring**: If you selected an issuer keyring previously, that keyring displays. If you did not select an issuer keyring previously, the default keyring displays. To change the keyring that is used as the issuer keyring, choose a different keyring from the drop-down menu.

b. **Hostname:** The hostname you put here is the hostname in the emulated certificate.

c. **Splash Text**: You are limited to a maximum of 200 characters. The splash text is added to the emulated certificate as a certificate extension.

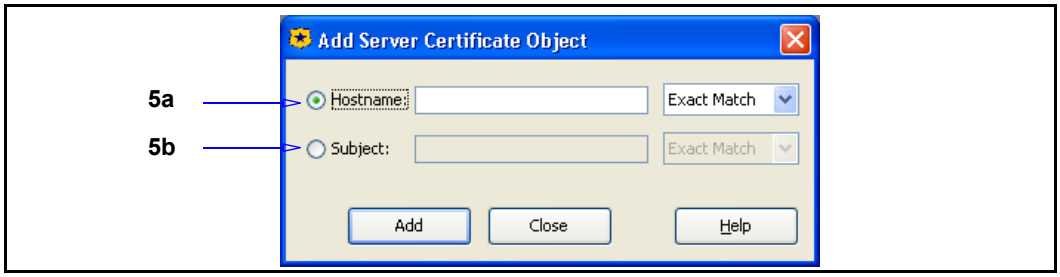
d. **Splash URL**: The splash URL is added to the emulated certificate as a certificate extension.

5. Click **OK** to save the changes.

You can use the **Disable SSL Intercept** object to disable HTTPS Intercept.

To intercept HTTPS requests to specific sites through VPM:

1. Select **Configuration > Policy > Visual Policy Manager** and launch VPM.
2. From the **Policy** drop-down menu, select **Add SSL Intercept Layer.**
3. In the **Destination** column, right-click **Set**; the **Set Destination Object** displays.
4. Click **New** and select **Server Certificate Validation.**



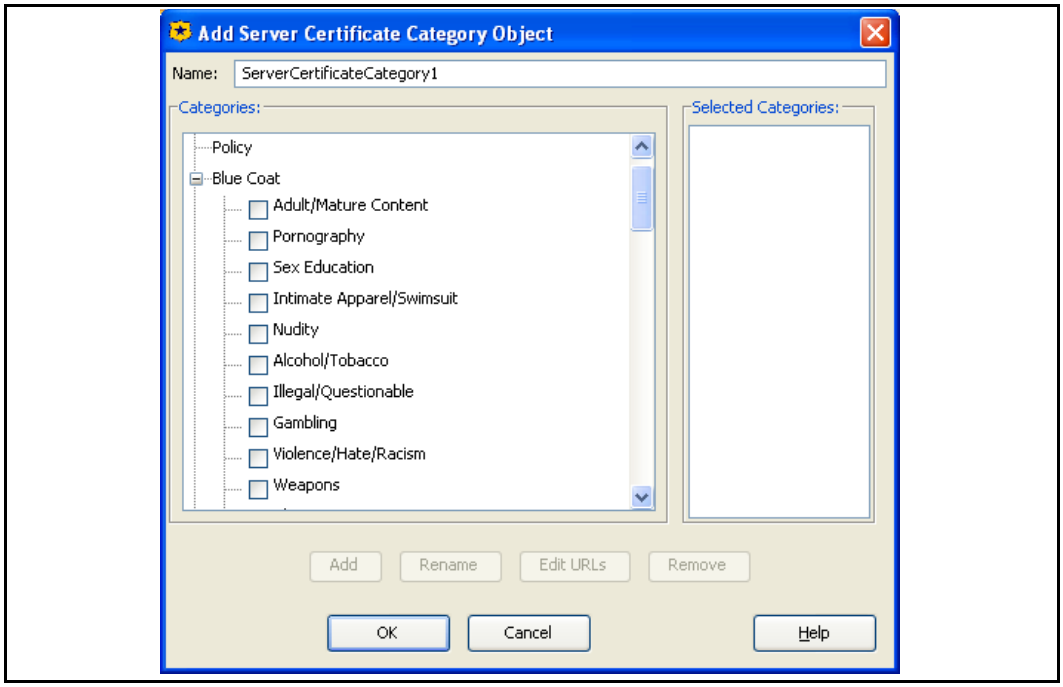
5. Fill in the fields as described below. You can only choose one field:

a. **Hostname**: This is the hostname of the server whose traffic you want to intercept. After entering the hostname, use the drop-down menu to specify **Exact Match**, **Contains**, **At Beginning**, **At End**, **Domain**, or **Regex**.

b. **Subject:** This is the subject field in the server's certificate. After you enter the subject, use the drop-down menu to specify **Exact Match**, **Contains**, **At Beginning**, **At End**, **Domain**, or **Regex**.

To categorize hostnames in server certificates through VPM:

1. While still in the Destination column of the **SSL Intercept** layer, right-click **Set**; the **Set Destination** object displays.
2. Click **New** and select the **Server Certificate Category** object. The **Add Server Certificate Category Object** displays. You can change the name in the top field if needed.



3. Select the categories. The categories you selected display in the right-hand column.
4. Click **OK**.

Using the SSL Access Layer

For a list of the conditions, properties, and actions that can be used in the SSL Access layer, see "CPL in the SSL Layer" on page 195.

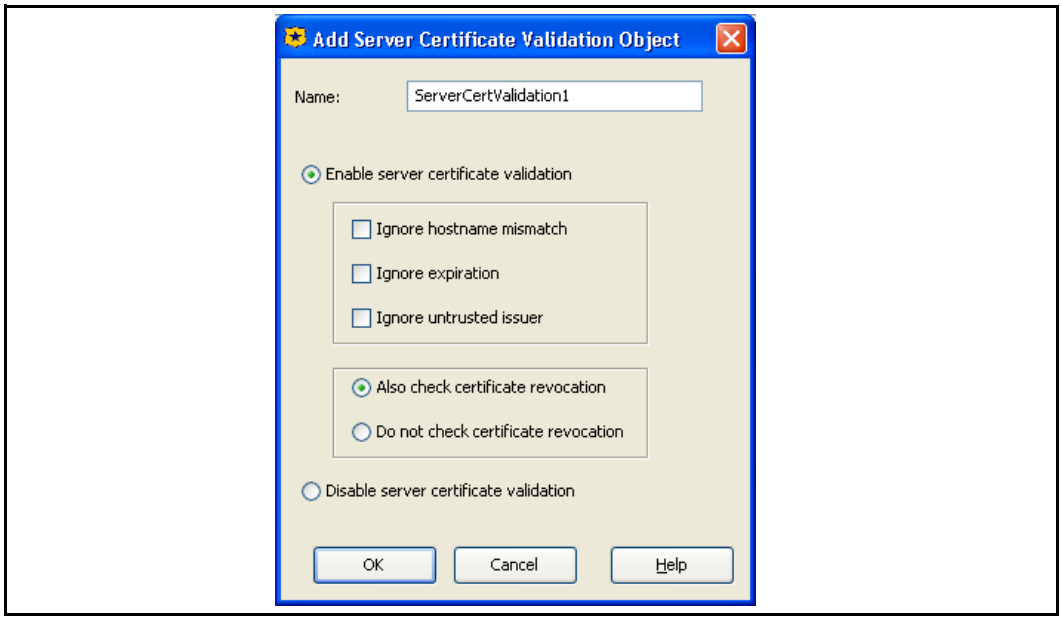
Note: For detailed instructions on using VPM, refer to *Volume 6: The Visual Policy Manager and Advanced Policy*.

To customize server certificate validation through VPM:

Note: The policy property `server.certificate.validate`, if set, overrides the `ssl-verify-server` command for either HTTP or for forwarding hosts.

---

1. Select **Configuration > Policy > Visual Policy Manager** and launch VPM.
2. From the **Policy** drop-down menu, select **Add SSL Access Layer.**
3. In the **Action** column, right-click **Set**; the **Set Action** object displays.
4. Click **New** and select **Set Server Certificate Validation** object.



5. By default, server certificate validation is enabled; to disable it, select **Disable server certificate validation** at the bottom of the dialog.

   If server certificate validation is enabled, you can determine behavior by selecting checkboxes to **Ignore a hostname mismatch**, **Ignore certificate expiration**, or **Ignore untrusted issuer.** These overrides mimic the overrides supported by most browsers.

   You can add server certificates to the ProxySG to allow proper validation. For more information, refer to *Volume 4: Securing the Blue Coat ProxySG Appliance.*

6. If you want to check the CA certificate revocation list (CRL) from a Certificate Authority, verify **Also check certification revocation** is selected. For information on using CRL, see "Checking CRLs" on page 178.

CPL in the SSL Intercept Layer

Note: VPM is much easier to use than CPL. All CPL gestures except the `ssl.forward_proxy.server_keyring` property, used only for troubleshooting, are also in VPM.

The following CPL gestures can be used in the SSL Intercept layer:

Note: No authentication-related triggers are allowed in the SSL Intercept layer.

Allowed Properties (allowed in the SSL Intercept layer only):

- `ssl.forward_proxy( )`
- `ssl.forward_proxy.splash_text( )`

- `ssl.forward_proxy.hostname( )`
- `ssl.forward_proxy.issuer_keyring ( )`
- `ssl.forward_proxy.server_keyring ( )`
- `ssl.forward_proxy.splash_url( )`
- `trace.destination( )`
- `trace.request( )`
- `trace.rules( )`
- `ssl.forward_proxy.server_keyring` (used for troubleshooting only)

Allowed Actions

- `log_message( )`
- `notify_email( )`
- `notify_snmp( )`
-

Allowed Conditions

- `category`
- `client.address`
- `client.host`
- `client.host.has_name`
- `client.protocol`
- `proxy.address`
- `proxy.card`
- `proxy.port`
- `server.certificate.hostname`
- `server.certificate.hostname.category`
- `server.certificate.subject`
- `server_url.*`
- `url.*`
-

An example of using CPL to intercept SSL traffic is:

```
;create list of servers to intercept
define condition server_intercept_list
  server.certificate.hostname.category=webmail
  server.certificate.hostname=porn.com
  server.certificate.hostname.category=gambling
  server.certificate.hostname.category=none
end condition server_intercept_list
<SSL-Intercept>
; value no means tunnel, value https means intercept as forward proxy
  condition=server_intercept_list ssl.forward_proxy(https)
  ssl.forward_proxy(no)
```

Note: For detailed instructions on using CPL, including detailed explanations of the gestures listed here, refer to *Volume 10: Content Policy Language Guide*

CPL in the SSL Layer

The following CPL gestures can be used in the SSL layer (called SSL Access layer in VPM):

Allowed Actions (allowed in the SSL layer only)

- `server.certificate.validate(yes | no)`
- `server.certificate.validate.check_revocation (local | no))`
- `server.certificate.validate.ignore (hostname_mismatch | expiration | untrusted_issuer)`
- `client.certificate.validate(yes | no)`
- `client.certificate.validate.check_revocation (local | no)`
- `client.certificate.require(yes)`

Allowed Conditions and Properties

- `client.connection.negotiated_ssl_version = (`*`condition`*`)`
- `client.certificate.common_name.regex = <regex>`
- `client.certificate.subject.dn = <X.500 DN>`
- `client.certificate.common_name[.exact|.substring | .prefix|.suffix] = <string>`
- `client.certificate.subject [.exact|.substring | .prefix|.suffix|.regex] = <string>`
- `client.certificate.subject.regex = <regex>`
- `server.certificate.hostname[.exact| .substring|.prefix|.suffix]=<string>`
- `server.certificate.hostname.regex= <regex>`
- `server.certificate.hostname. category = <category_list>`
- `server.certificate.hostname.category =! <exclusion_category_list> (`*`condition`*`)`
- `server.connection.negotiated_cipher =`
- `server.connection.negotiated_cipher.strength = low | medium | high | export`
- `ssl.proxy_mode=`
- `client.protocol= tunneled=`

Note: For detailed instructions on using CPL, including detailed explanations of the gestures listed here, refer to *Volume 10: Content Policy Language Guide.*

Notes

Note: Pipelining configuration for HTTP is ignored for HTTPS requests intercepted by the SSL proxy. When the SSL proxy intercepts an HTTPS request, and the response is an HTML page with embedded images, the embedded images are not pre-fetched by the ProxySG.

- If the ProxySG and the origin content server cannot agree on a common cipher suite for intercepted connections, the connection is aborted.

- Server-Gated Cryptography and step-up certificates are treated just as regular certificates; special extensions present in these certificates are not be copied into the emulated certificate. Clients relying on SGC / step-up certificates continue using weaker ciphers between the client and the ProxySG when the SSL proxy intercepts the traffic.

Section C: Viewing SSL Statistics

SSL History Statistics

The **Statistics > Protocol details > SSL History** tabs (**Unintercepted SSL Data**, **Unintercepted SSL Clients**, **Unintercepted SSL Bytes**) provide various useful statistics for unintercepted SSL traffic.

Note: Some SSL statistics (SSL client connections and total bytes sent and received over a period of time) can only be viewed through the Management Console (see "Unintercepted SSL Data" on page 197 and "Unintercepted SSL Clients" on page 198).

Unintercepted SSL Data

The **Unintercepted SSL Data** tab on the Management Console displays SSL statistics.

The following table details the statistics provided through the **Unintercepted SSL Data** tab.

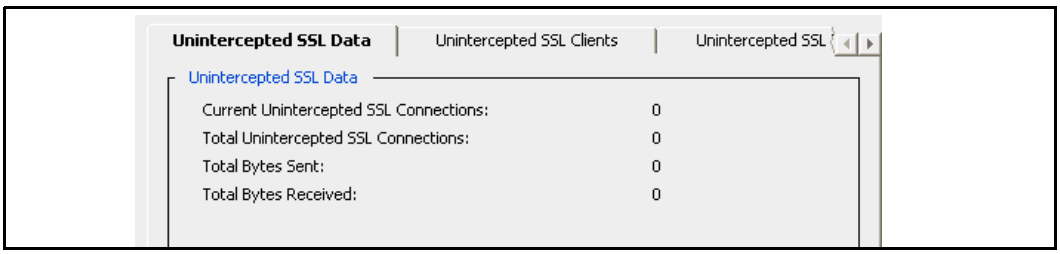
Table 12–1 Unintercepted SSL Data Statistics

Status	Description
Current Unintercepted SSL Sessions	The current number of unintercepted SSL client connections.
Total Unintercepted SSL Sessions	The cumulative number of unintercepted SSL client connections since the ProxySG was last rebooted.
Total Bytes Sent	The total number of unintercepted bytes sent.
Total Bytes Received	The total number of unintercepted bytes received.

To view unintercepted SSL data statistics:

From the Management Console, select **Statistics > Protocol Details > SSL History > Unintercepted SSL Data.**

The default view shows all unintercepted SSL data.

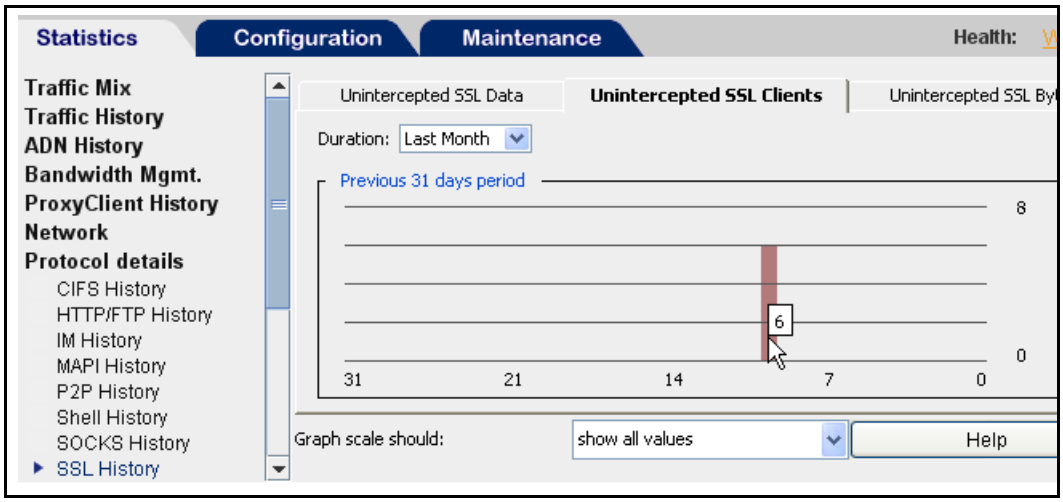


Unintercepted SSL Clients

The **Unintercepted SSL Clients** tab displays dynamic graphical statistics for connections received in the last 60-minute, 24-hour, or 30-day period.

To view SSL client unintercepted statistics:

1. From the Management Console, select **Statistics > Protocol Details > SSL History > Unintercepted SSL Clients**.



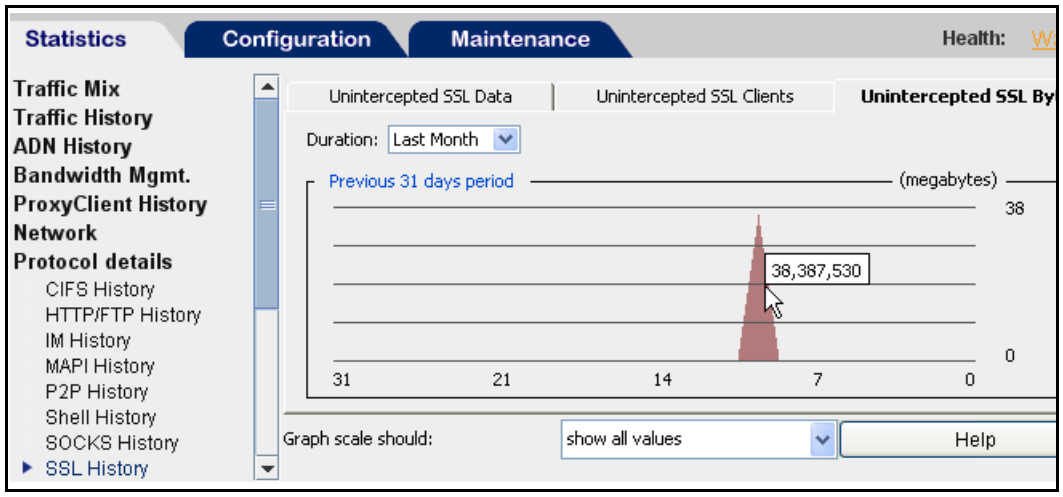
2. Select a time period for the graph from the **Duration**: drop-down list. The default is Last Week.
3. (Optional) To set the graph scale to a different value, select a value from the **Graph scale should** drop-down list.

Unintercepted SSL Bytes

The **Unintercepted SSL Bytes** tab displays dynamic graphical statistics for bytes received in the last 60-minute, 24-hour, or 30-day period.

To view unintercepted SSL byte statistics:

1. From the Management Console, select **Statistics > Protocol Details > SSL History > Unintercepted SSL Bytes**.



2. Select the **Duration**: for the graph from the drop-down list. The default is Last week.
3. (Optional) To set the graph scale to a different value, select a value from the **Graph scale should** drop-down list.

Section D: Advanced Topics

If you use OpenSSL or Active Directory, you can follow the procedures below to manage your certificates.

For OpenSSL, see "Creating an Intermediate CA using OpenSSL" on page 200; if using Active Directory, see "Creating an Intermediate CA using Microsoft Server 2003 (Active Directory)" on page 203.

Creating an Intermediate CA using OpenSSL

This section describes the certificate management when creating an intermediate CA using OpenSSL.

The overall steps are:

- "Installing OpenSSL" on page 200
- "Creating a Root Certificate" on page 200
- "Modifying the OpenSSL.cnf File" on page 201
- "Signing the ProxySG CSR" on page 202
- "Importing the Certificate into the ProxySG" on page 202
- "Testing the Configuration" on page 202

Various OpenSSL distributions can be found at `http://www.openssl.org`.

Installing OpenSSL

After OpenSSL is installed, you must edit the `openssl.cnf` file and ensure the pathnames are correct. By default root certificates are located under `./PEM/DemoCA`; generated certificates are located under `/certs`.

Creating a Root Certificate

In order to create a root Certificate Authority (CA) certificate, complete the following steps.

Note: The key and certificate in this example is located at `./bin/PEM/demoCA/private/`.

---

1. Open a MS-DOS window, and enter:

   ```
   openssl req -new -x509 -keyout
   c:\resources\ssl\openssl\bin\PEM\demoCA\private\
   cakey.pem -out
   c:\resources\ssl\openssl\bin\PEM\demoCA\private\CAcert.pem
   ```

   where the root directory for openssl is: `\resources\ssl\openssl`

```
openssl req -new -x509 -keyout
c:\resources\ssl\openssl\bin\PEM\demoCA\private\cakey.pem -out
c:\resources\ssl\openssl\bin\PEM\demoCA\private\CAcert.pem
Using configuration from C:\Resources\SSL\OpenSSL\bin\openssl.cnf
Loading 'screen' into random state - done
Generating a 1024 bit RSA private key
.....................................+++++
.................................................+++++
writing new private key to
'c:\resources\ssl\openssl\bin\PEM\demoCA\private\cakey.pem'
Enter PEM pass phrase:
```

2. Type any string more than four characters for the PEM pass phrase.
3. Enter the certificate parameters, such as country name, common name that are required for a Certificate Signing Request (CSR).

   The private key and root CA are now located under the directory `./PEM/DemoCA/private`
4. Create a ProxySG keyring.
   a. From the Management Console, select **Configuration > SSL > Keyrings.**
   b. Click **Create**; fill in the fields as appropriate.
   c. Click **OK**.
5. Create a CSR on the ProxySG.
   a. From the Management Console, select **Configuration > SSL > Keyrings.**
   b. Highlight the keyring you just created; click **Edit/View.**
   c. In the Certificate Signing Request pane, click **Create** and fill in the fields as appropriate.

   Note: Detailed instructions on creating a keyring and a CSR are in *Volume 4: Securing the Blue Coat ProxySG Appliance*. They can also be found in the online help.
6. Paste the contents of the CSR into a text file called `new.pem` located in the `./bin` directory.

Modifying the OpenSSL.cnf File

Modify the `openssl.cnf` file to import the openSSL root CA into your browser. If you do not do this step, you must import he ProxySG certificate into the browser.

1. In the openssl.cnf file, look for the string `basicConstraints=CA`, and set it to `TRUE`.

   `basicConstraints=CA:TRUE`
2. Save the `openSSL.cnf` file.

Section D: Advanced Topics

Signing the ProxySG CSR

Open a MS-DOS window and enter:

```
openssl ca -policy policy_anything -out newcert.pem -in new.pem
```

The output is:

```
Using configuration from C:\Resources\SSL\OpenSSL\bin\openssl.cnf
Enter PEM pass phrase:
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName           :PRINTABLE:'FR'
stateOrProvinceName   :PRINTABLE:'Paris'
localityName          :PRINTABLE:'Paris'
organizationName      :PRINTABLE:'BlueCoat'
organizationalUnitName:PRINTABLE:'Security Team'
commonName            :PRINTABLE:'ProxySG.bluecoat.com'
emailAddress          :IA5STRING:'support@bc.com'
Certificate is to be certified until Sep 27 13:29:09 2006 GMT (365
days)
Sign the certificate? [y/n]:y
1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

This signs the certificate; it can then be imported into the ProxySG.

Importing the Certificate into the ProxySG

1. Open the file `newcert.pem` in a text editor.
2. Select **Management Console > Configuration > SSL > SSL Keyrings**.
3. Selecting the keyring used for SSL interception; click **Edit/View.**
4. Paste in the contents of the `newcert.pem` file.
5. Import the contents of the `newcert.pem` file into the CA Certificates list.
   a. From the Management Console, select **Configuration > SSL > CA Certificates**.
   b. Click **Import**; enter the certificate name in the CA Cert Name field.
   c. Paste the certificate, being sure to include the `-----BEGIN CERTIFICATE----` and the `----END CERTIFICATE-----` statements in the `./bin/PEM/demoCA/private/CAcert` file.
   d. Click **OK**.

> **Note:** Detailed instructions on importing a CA certificate are in Chapter 9: "Configuing and Managing an HTTPS Reverse Proxy Service" on page 151.

Testing the Configuration

Import the root CA into your browser and construct an SSL interception policy.

Note: Detailed instructions on constructing an SSL interception policy are in Section B: "Configuring SSL Rules through Policy" on page 190.

You should not be prompted for any certificate warning.

Creating an Intermediate CA using Microsoft Server 2003 (Active Directory)

This section describes certificate management when creating an intermediate CA using Active Directory.

Before you begin:

- Verify the Windows 2003 system is an Active Directory server.
- Make sure IIS is installed.
- Install the "Certificate Services" through the control panel
- Select the mode to be Enterprise root CA.

All certificate management is done through the browser using the following URL:

```
http://@ip_server/CertSrv
```

For information on the following tasks, see:

- "To install the root CA onto the browser:" on page 203
- "To create a ProxySG keyring and certificate signing request:" on page 203
- "To sign the ProxySG CSR:" on page 204
- "To import the certificate onto the ProxySG:" on page 204
- "To test the configuration:" on page 204

To install the root CA onto the browser:

1. Connect to `http://@ip_server/CertSrv`.
2. Click **Download a CA Certificate**.
3. Click **Install this CA Certificate chain**.

This installs the root CA onto the browser.

To create a ProxySG keyring and certificate signing request:

1. From the Management Console, select **SSL > Keyrings**.
2. Create a new keyring. For detailed instructions on creating a new keyring, refer to *Volume 4: Securing the Blue Coat ProxySG Appliance*.
3. Create a Certificate Signing Request (CSR). For detailed instructions on creating a CSR, refer to *Volume 4: Securing the Blue Coat ProxySG Appliance*.
4. Click **OK**.

To sign the ProxySG CSR:

1. Connect to `http://@ip_server/CertSrv`.
2. Select the option **Request a certificate**.
3. Select **Submit an advanced certificate request** and then **Submit a certificate request by using a base 64 encoded …**
4. Paste the contents of the CSR.
5. Select the Certificate Template **Subordinate Certification Authority.**

   If this template does not exist, connect to the certificate manager tool on the Active Directory server and add the template.
6. Click on **Submit**.
7. Download the certificate (not the chain) as **Base 64 encoded**.
8. Save this file on the workstation as `newcert.pem`.

To import the certificate onto the ProxySG:

1. Open the file `newcert.pem` in a text editor.
2. In the Management Console, select **Configuration > SSL > SSL Keyrings**.
3. Select the keyring that has the CSR created; click **Edit/View.**

   Note: Make sure this keyring is used as the issuer keyring for emulated certificates. Use policy or the SSL intercept setting in the Management Console or the CLI.
4. Paste the contents of the `newcert.pem` file.
5. Import the contents of the `newcert.pem` file into the CA Certificates list.
   a. From the Management Console, select **Configuration > SSL > CA Certificates**.
   b. Click **Import**; enter the certificate name in the **CA Cert Name** field.
   c. Paste the certificate, being sure to include the `-----BEGIN CERTIFICATE----` and the `----END CERTIFICATE-----` statements in the `./bin/PEM/demoCA/private/CAcert` file.
   d. Click **OK**.

   Note: Detailed instructions on importing a CA certificate are in Chapter 9: "Configuing and Managing an HTTPS Reverse Proxy Service" on page 151.

To test the configuration:

Import the root CA into your browser and construct a SSL interception policy.

Note: Detailed instructions on constructing an SSL interception policy are in Section B: "Configuring SSL Rules through Policy" on page 190.

You should not be prompted for any certificate warning.

Chapter 13: Managing the TCP Tunneling Proxy

This chapter discusses managing traffic through the ProxySG TCP Tunneling Proxy. Tunneling, or port forwarding, is a way to forward TCP traffic. Any application protocol running over TCP can be tunneled using this service. Client-server applications carry out any authentication procedures just as they do when TCP tunneling is not involved.

SGOS uses a `tcp://` scheme for TCP-tunnel transactions instead of HTTPS because SGOS does not actually know that it is HTTPS that is being tunneled.

You can use ADN optimization in conjunction with TCP tunnels to compress and accelerate the tunneled traffic. Both explicit and transparent TCP tunneling are supported. Which one you use depends on your needs.

- *Explicit* TCP tunneling allows connections to one of the ProxySG's IP addresses.
- *Transparent* TCP tunneling allows connections to any IP address other than those belonging to the ProxySG. TCP tunneling in transparent mode supports categorization as well as blocking of destination IP address, port, host, and domain.

Note: The TCP-Tunnel service does not support content filtering with Websense offbox or ICAP.

Topics in this Chapter

This chapter includes information about the following topics:

- "TCP-Tunnel Proxy Services Supported" on page 207
- "Creating or Editing a TCP-Tunnel Proxy Service" on page 208

TCP-Tunnel Proxy Services Supported

A number of proxy services are supported with the TCP-Tunnel proxy. For the most current list, see Table 3–5: "Proxy Name and Listeners" on page 55.

In addition, the default proxy service (which listens on all ports not assigned to other services), uses the TCP-Tunnel proxy. The default proxy service has only one listener; its action can be set to bypass or intercept. No new listeners can be added to the default proxy service, and the default listener and service cannot be deleted. Service attributes can be changed.

To keep the ProxySG from interfering with unassigned traffic, set the behavior to bypass.

An access log entry is available for every TCP tunnel connection.

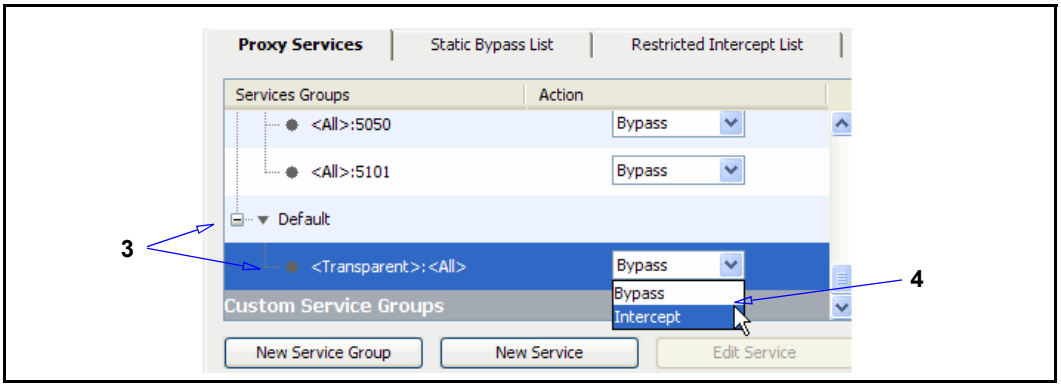
Configuring the TCP-Tunnel Proxy Service Options

This section describes how to change the default service options and add new services.

Changing the TCP-Tunnel Proxy Service to Intercept All IP Addresses on All Unattended Ports

The service is configured to listen to all IP addresses, but is set in Bypass mode. The following procedure describes how to change the service to Intercept mode. To configure the TCP-Tunnel proxy to intercept traffic:

1. From the Management Console, select **Configuration > Services > Proxy Services**.
2. To edit an existing TCP-Tunnel proxy service, highlight the service and click **Edit**. To create a new proxy service, click **New**.

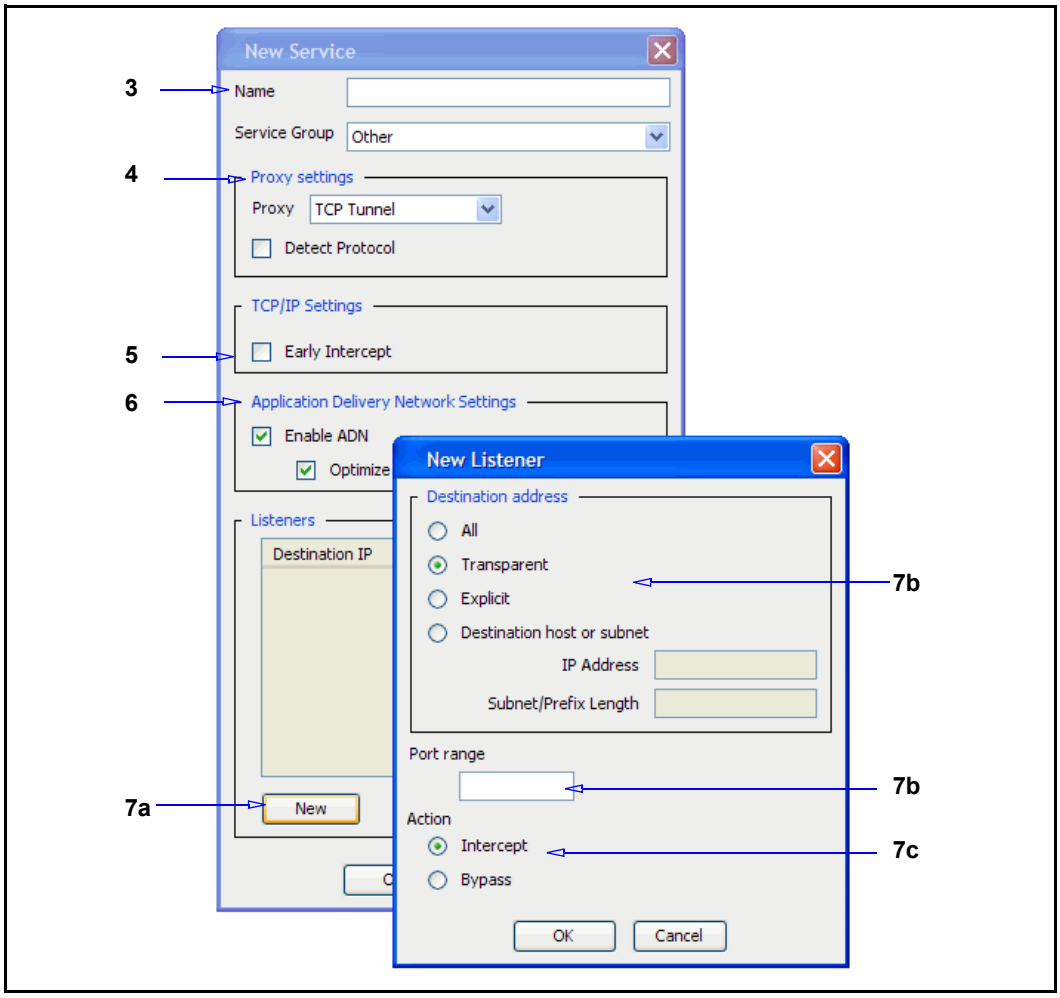


3. Scroll to the **Other** service group and click to expand the list; scroll down to the **Default** service and select it to expand the list; select **<Transparent>:<All>**.
4. Notice the **Action** for each default service is **Bypass**. Select **Intercept** from the drop-down list.
5. Click **Apply**.

Creating or Editing a TCP-Tunnel Proxy Service

This procedure

1. Select **Configuration > Services > Proxy Services**.
2. To edit a TCP-Tunnel proxy service, highlight the service and click **Edit**. To create a new proxy service, click **New**.



3. If you are creating a new TCP-Tunnel proxy service, enter a meaningful name in the **Name** field.
4. Configure Proxy Settings options:
   a. In the **Proxy settings** field, select TCP Tunnel from the drop-down menu.
   b. Select the **Detect Protocol** checkbox to automatically detect the protocol being used. This breaks connections that do not have the client send information first, but expect the server to respond on connection. It also can add significant delay if the client does not send specific information, and only after timing out does it treat the traffic as unknown.
5. **Early intercept**: Controls whether the proxy responds to client TCP connection requests before connecting to the upstream server. When early intercept is disabled, the proxy delays responding to the client until after it has attempted to contact the server.
6. Configure ADN options:
   a. **Enable ADN**. Select this option if you want this service to use ADN.

Enabling ADN does not guarantee the connections are accelerated by ADN. The actual enable decision is determined by ADN routing (for explicit deployment) and network setup (for transparent deployment).

b. The **Optimize Bandwidth** option is selected by default if you enabled ADN optimization during initial configuration. Clear the option if you are not configuring ADN optimization.

7. Create a new listener:
   a. Click **New.**
   b. Define the IP address option: explicit or the specified address.
   c. In the **Port Range** field, enter the ports on which the service should listen. The default ports for each service are listed in Table 3–5, "Proxy Name and Listeners" on page 55.
   d. Select the default action for the service: **Bypass** tells the service to ignore any traffic. **Intercept** configures the service to intercept the traffic that is being proxied.

      If you selected **Optimize all other TCP traffic** during initial configuration, all listeners in services that use the TCP-Tunnel proxy intercept traffic. If you did not select **Optimize all other TCP traffic,** TCP-Tunnel listeners bypass all traffic by default.
   e. Click **OK** to close the listener dialog.
8. Click **OK** to close the service dialog.
9. Click **Apply.**

Related CLI Syntax to Create/Edit a Tunneling Proxy Service

- To enter configuration mode:

```
SGOS#(config) proxy-services
SGOS#(config proxy-services) create tcp-tunnel service-name
SGOS#(config proxy-services) edit service-name
```

- The following subcommands are available:

```
SGOS#(config service-name) add {transparent | explicit | all |
ip_address | ip_address/subnet-mask} {port | first_port-last_port}
[intercept | bypass]
SGOS#(config service-name) attribute {adn-optimize {enable | disable}|
detect-protocol {enable | disable}| early-intercept {enable |
disable}| use-adn {enable | disable}}
SGOS#(config service-name) bypass {transparent | explicit | all |
ip_address | ip_address/subnet-mask} {port | first_port-last_port}
SGOS#(config service-name) exit
SGOS#(config service-name) intercept {transparent | explicit | all |
ip_address | ip_address/subnet-mask} {port | first_port-last_port}
SGOS#(config service-name) remove {transparent | explicit | all |
ip_address | ip_address/subnet-mask} {port | first_port-last_port}
SGOS#(config service-name) view
```

If you created a transparent TCP-Tunnel service, the procedure is complete. If you created an explicit TCP-Tunnel service, you must configure a forwarding destination port.

To configure a forwarding destination port:

1. Create a forwarding destination port, where the ProxySG directs traffic.

```
SGOS#(config proxy-services tcp-tunnel) exit
SGOS#(config proxy-services) exit
SGOS#(config) forwarding
SGOS#(config forwarding) create host_alias ip_address tcp=port
```

2. (Optional) View the results:

```
SGOS#(config forwarding) view
Forwarding Groups: (* = host unresolved)
No forwarding groups defined.
Individual Hosts: (* = host unresolved)
Host_Alias 10.25.36.47 tcp=port_number
```

Appendix A: Explicit and Transparent Proxy

Whether you select explicit or transparent proxy deployment is determined by factors such as network configuration, number of desktops, desired user experience, and desired authentication approach.

Note: While you must configure proxying to do authentication, verify the proxy is configured correctly and is functioning before adding authentication to the mix. Many network or other configuration problems can appear similar to authentication errors.

About the Explicit Proxy

In an explicit proxy configuration, the client (browser) is explicitly configured to use a proxy server. The browser is given the IP address and port number of the proxy service (the ProxySG). It is also possible to configure the browser to download the proxy settings from a Web server. This is called a Proxy Auto-Configuration (PAC) file. When a user makes a request, the browser connects to the proxy service and sends the request. Because the browser knows it is talking to a proxy, the browser provides the proxy server with the destination server.

The proxy service accepts the explicit connection to it, and fetches the request from the browser. The request identifies the desired origin content server (OCS) and the resource on that server. The proxy service uses this information to contact the OCS if necessary.

The disadvantage to explicit proxy is that each desktop must be properly configured to use the proxy, which might not be feasible in a large organization.

Note: Explicit proxy allows a redundant configuration using IP address failover among a cluster of machines. For information on creating a redundant configuration for failover, refer to *Volume 5: Advanced Networking*.

About the Transparent Proxy

When transparent proxy is enabled, the client (browser) does not know the traffic is being processed by a machine other than the OCS. The browser believes it is talking to the OCS, so the request is formatted for the OCS and the proxy determines for itself the destination server based on information in the request, such as the destination IP address in the packet, or the `Host:` header in the request.

To enable the ProxySG to intercept traffic sent to it, you must create a service and define it as transparent. The service is configured to intercept traffic for a specified port, or for all IP addresses on that port. A transparent HTTP proxy, for example, typically intercepts all traffic on port 80 (all IP addresses).

To make sure that the appropriate traffic is directed to the ProxySG, deploy hardware such as a Layer-4 switch or a WCCP router, or the ProxySG's software bridge that can redirect selected traffic to the appliance. Traffic redirection is managed through polices you create on the redirection device.

For detailed information on explicit proxies, continue with the next section; for detailed information on transparent proxies, continue with "Transparent Proxies" on page 215.

For information on creating an explicit proxy server, regardless of proxy type, continue with "Creating an Explicit Proxy Server" on page 214.

Creating an Explicit Proxy Server

If your network does not use transparent proxy, clients on the network must configure their browsers to use either an explicit proxy server or a Proxy Auto-Configuration (PAC) file.

Two PAC files ship with the ProxySG:

- default PAC file
- accelerated PAC file

They can be accessed at:

- `https://ProxySG_IP_Address:8082/accelerated_pac_base.pac`
- `https://ProxySG_IP_Address:8082/proxy_pac_file`

Note: Only the `accelerated_pac_base.pac` file can be edited. Any text editor can be used.

The ProxySG generates client instructions that describe how to configure Microsoft Internet Explorer, Netscape Communicator, and Firefox based on instructions selected by the ProxySG administrator. You can configure client instructions for each network adapter in the ProxySG with the **Configuration > Network > Adapters > Interface > Settings** button.

After selecting client instructions, the ProxySG administrator directs clients to go to the ProxySG home page and follow the instructions in the Browser Configuration section. The ProxySG detects the browser installed on the client and displays the appropriate instructions.

Using the ProxySG as an Explicit Proxy

To use the ProxySG as an explicit proxy and use services such as SOCKS or FTP, you must provide custom instructions to clients instructing them how to configure their browsers to use the ProxySG as a proxy server.

This is a two-step process, requiring that you add the proxy IP address to the browser and also instruct the ProxySG which adapter interface uses the proxy IP address.

Before the proxy can be used, you must:

- Configure the proxy server.

- Enable the explicit proxy (whether a service or a server).

The browsers described here are Internet Explorer 6.0 and Firefox 1.5. If you have different browsers or different versions of Internet Explorer or Firefox, refer to the vendor documentation for information on configuring proxies.

From Internet Explorer:

1. Select **Tools > Internet Options > Connections > LAN Settings**.
2. Click **Use a proxy server.**
3. Enter the IP address and port number for the proxy, or click **Advanced** to set proxy server IP addresses and port numbers for services such as HTTP, FTP, and SOCKS. (Configure HTTPS through the **Secure** field.)
4. Click **OK** to exit the **Advanced Settings** tab, then continue to click **OK** until you exit the **Tools** menu.

From Firefox:

1. Select **Tools > Options > Genera l> Connection Settings**.
2. Click **Manual proxy configuration.**
3. Enter proxy server IP addresses and port numbers for services such as HTTP, FTP, SOCKS, and SSL.
4. Click **OK**; click **OK** again.

Configuring Adapter Proxy Settings

Once the explicit proxy is configured on the browser, decide which adapter interfaces listen for which service. Each adapter interface can listen for only one IP address; you can configure multiple proxies on one ProxySG using the same IP address.

To provide configuration instructions on the ProxySG:

1. Select **Configuration > Network > Adapters.**
2. In the Adapter pane, select the adapter you want to use. If an adapter does not exist, the Adapter pane displays the word Empty.
3. In the Interface pane, select the correct interface. Click **Settings**.
4. Select **Using a proxy.**
5. Click **OK** to close the Settings dialog.

Relevant CLI Syntax to Configure Adapter Proxy Settings

```
SGOS#(config) interface fast-ethernet interface_#
```

Transparent Proxies

A transparent proxy can be configured several ways:

- Through hardware: See "Configuring Transparent Proxy Hardware" on page 216.

- Through bridging: "Bridging" on page 216.
- Through using the ProxySG as a gateway: See "Configuring IP Forwarding" on page 217.

In addition to the transparent proxy configuration, you must create a proxy service for the transparent proxy and enable the service. At this time, you can also set other attributes for the service, including the destination IP address and port range. For information on creating or editing a proxy service for transparent configuration, see Chapter 3: "About Proxy Services and Proxies" on page 27.

Configuring Transparent Proxy Hardware

For transparent proxy to work, you must use one of the following:

- A bridge, either hardware or software
- Layer-4 switch
- WCCP

Bridging

Network bridging through the ProxySG provides transparent proxy pass-through and failover support. This functionality allows ProxySGs to be deployed in environments where L4 switches and WCCP-capable routers are not feasible options.

The ProxySG provides bridging functionality by two methods:

- Software—A software, or *dynamic*, bridge is constructed using a set of installed interfaces. Within each logical bridge, interfaces can be assigned or removed. Note that the adapters must of the same type. Although the software does not restrict you from configuring bridges with adapters of different types (10/100 or GIGE), the resultant behavior is unpredictable.

  To set up a software bridge, refer to *Volume 1: Getting Started*.

- Hardware—The Blue Coat Pass-Through card is a 10/100 dual interface Ethernet device that enables a bridge, using its two adapters, so that packets can be forwarded across it. However, if the system crashes, the Pass-Through card becomes a network: the two Ethernet cables are connected so that traffic can continue to pass through without restriction.

  When the Pass-Through card is installed on the ProxySG, a bridge is automatically created and traffic going through the bridge is intercepted according to the proxy-service setting. Note that:

  - Forwarding traffic behavior: By default, the bridge forwards packets that are not to be intercepted.
  - Proxy request behavior: Requests are proxied on either adapter, so if you connect one side of the bridge to your Internet connection, there might be a number of issues.

Configuring a Layer-4 Switch

In transparent proxy acceleration, as traffic is sent to the origin content server, any traffic sent on port 80 is redirected to the ProxySG by the Layer 4 switch. The benefits to using a Layer 4 switch include:

- Built-in failover protection. In a multi-ProxySG setup, if one fails, the Layer 4 switch can route to the next ProxySG.
- Request partitioning based on IP address instead of on HTTP transparent proxying. (This feature is not available on all Layer 4 switches.)
- ProxySG bypass prevention. You can configure a Layer 4 device to always go through the ProxySG even for requests to a specific IP address.
- ProxySG bypass enabling. You can configure a Layer 4 device to never go through the ProxySG.

For information on configuring a layer-4 switch, refer to the manufacturer's documentation.

Configuring a WCCP-Capable Router

WCCP is a Cisco®-developed protocol that allows you to establish redirection of the traffic that flows through routers.

The main benefits of using WCCP are:

- Scalability — With no reconfiguration overhead, redirected traffic can be automatically distributed to up to 32 ProxySGs.
- Redirection safeguards — If no ProxySGs are available, redirection stops and the router forwards traffic to the original destination address.

For information on using WCCP with a ProxySG, refer to *Volume 5: Advanced Networking*.

Configuring IP Forwarding

IP Forwarding is a special type of transparent proxy. The ProxySG is configured to act as a gateway and is configured so that if a packet is addressed to the ProxySG adapter, but not its IP address, the packet is forwarded toward the final destination. If IP forwarding is disabled, the packet is rejected as being mis-addressed.

By default, IP forwarding is disabled to maintain a secure network.

Important: When IP forwarding is enabled, be aware that all ProxySG ports are open and all the traffic coming through them is not subjected to policy, with the exception of the ports that have explicitly defined through the **Configuration > Services > Proxy Services** tab.

To enable IP forwarding:

1. Select **Configuration > Network > Routing > Gateways.**
2. Select the **Enable IP forwarding** checkbox at the bottom of the pane.
3. Click **OK**; click **Apply.**

Related CLI Syntax to Enable IP Forwarding

```
SGOS#(config) tcp-ip ip-forwarding enable
```

Glossary

A

access control list—Allows or denies specific IP addresses access to a server.

access log—A list of all the requests sent to a ProxySG. You can read an access log using any of the popular log-reporting programs. When a client uses HTTP streaming, the streaming entry goes to the same access log.

account—A named entity that has purchased the ProxySG or the Entitlements from Blue Coat.

activation code—A string of approximately 10 characters that is generated and mailed to customers when they purchase the ProxySG.

active content stripping—Provides a way to identify potentially dangerous mobile or active content and scripts, and strip them out of a response.

active content types—Used in the Visual Policy Manager. Referring to Web Access policies, you can create and name lists of active content types to be stripped from Web pages. You have the additional option of specifying a customized message to be displayed to the user

administration access policy—A policy layer that determines who can access the ProxySG to perform administrative tasks.

administration authentication policy—A policy layer that determines how administrators accessing the ProxySG must authenticate.

AJAX—Acronym for Asynchronous JavaScript and XML, the technology used for live updating of Web objects without having to reload the entire page.

Application Delivery Network (ADN)—A WAN that has been optimized for acceleration and compression by Blue Coat. This network can also be secured through the use of appliance certificates. An ADN network is composed of an ADN manager and backup ADN manager, ADN nodes, and a network configuration that matches the environment.

ADN backup manager—Takes over for the ADN manager in the event it becomes unavailable. See *ADN manager*.

ADN manager—Responsible for publishing the routing table to SG Clients (and to other ProxySG appliances).

ADN optimize attribute—Controls whether to optimize bandwidth usage when connecting upstream using an ADN tunnel.

A record—The central records of DNS, which link a domain or subdomain to an IP address. An A record can correspond to a single IP address or many IP addresses.

asx rewrite—Allows you to rewrite URLs and then direct a client's subsequent request to the new URL. One of the main applications of ASX file rewrites is to provide explicit proxy-like support for Windows Media Player 6.4, which cannot set explicit proxy mode for protocols other than HTTP.

audit—A log that provides a record of who accessed what and how.

authenticate-401 attribute—All transparent and explicit requests received on the port always use transparent authentication (cookie or IP, depending on the configuration). This is especially useful to force transparent proxy authentication in some proxy-chaining scenarios

authenticated content—Cached content that requires authentication at the origin content server (OCS). Supported authentication types for cached data include basic authentication and IWA (or NTLM).

authentication—Allows you to verify the identity of a user. In its simplest form, this is done through usernames and passwords. Much more stringent authentication can be employed using digital certificates that have been issued and verified by a Certificate Authority. *See also* basic authentication, proxy authentication, and SSL authentication.

authentication realm—Authenticates and authorizes users to access SG services using either explicit proxy or transparent proxy mode. These realms integrate third-party vendors, such as LDAP, Windows, and Novell, with the Blue Coat operating system.

authorization—The permissions given to an authenticated user.

B

bandwidth—The amount of data you can send through a network or modem connection, usually measured in bits per second (bps).

bandwidth class—A defined unit of bandwidth allocation.

bandwidth class hierarchy—A gouping of bandwidth classes into a tree structure that specifies the relationship among different classes. You create a hierarchy by creating at least one parent class and assigning other classes as its children.

bandwidth gain—Bandwidth gain is a calculation of the savings that occur when bandwidth is not consumed as a result of some form of optimization.

For example, bandwidth gain for active sessions is calculated by subtracting the number of client bytes from the number of server bytes and dividing the result by the number of server bytes.

(Client Bytes - Server Bytes) / Server Bytes

bandwidth management—Classify, control, and, if needed, limit the amount of bandwidth used by network traffic flowing in or out of a ProxySG.

basic authentication—The standard authentication for communicating with the target as identified in the URL.

BCAAA—Blue Coat Authentication and Authorization Agent. Allows SGOS 5.x to manage authentication and authorization for IWA, CA eTrust SiteMinder realms, Oracle COREid, Novell, and Windows realms. The agent is installed and configured separately from SGOS 5.x and is available from the Blue Coat Web site.

BCLP—Blue Coat Licensing Portal.

byte-range support—The ability of the ProxySG to respond to byte-range requests (requests with a `Range:` HTTP header).

C

cache—An "object store," either hardware or software, that stores information (objects) for later retrieval. The first time the object is requested, it is stored, making subsequent requests for the same information much faster.

A cache helps reduce the response time and network bandwidth consumption on future, equivalent requests. The ProxySG serves as a cache by storing content from many users to minimize response time and prevent extraneous network traffic.

cache control—Allows you to configure which content the ProxySG stores.

cache efficiency—A tab found on the Statistics pages of the Management Console that shows the percent of objects served from cache, the percent loaded from the network, and the percent that were non-cacheable.

cache hit—Occurs when the ProxySG receives a request for an object and can serve the request from the cache without a trip to the origin server.

cache miss—Occurs when the ProxySG receives a request for an object that is not in the cache. The ProxySG must then fetch the requested object from the origin server.

cache object—Cache contents includes all objects currently stored by the ProxySG. Cache objects are not cleared when the ProxySG is powered off.

Certificate Authority (CA)—A trusted, third-party organization or company that issues digital certificates used to create digital signatures and public key / private key pairs. The role of the CA is to guarantee that the individuals or company representatives who are granted a unique certificate are who they claim to be.

child class (bandwidth gain)—The child of a parent class is dependent on that parent class for available bandwidth (they share the bandwidth in proportion to their minimum / maximum bandwidth values and priority levels). A child class with siblings (classes with the same parent class) shares bandwidth with those siblings in the same manner.

cipher suite—Specifies the algorithms used to secure an SSL connection. When a client makes an SSL connection to a server, it sends a list of the cipher suites that it supports.

client consent certificates—A certificate that indicates acceptance or denial of consent to decrypt an end user's HTTPS request.

client-side transparency—A way of replacing the ProxySG IP address with the Web server IP address for all port 80 traffic destined to go to the client. This effectively conceals the ProxySG address from the client and conceals the identity of the client from the Web server.

concentrator—A ProxySG, usually located in a data center, that provides access to data center resources, such as file servers.

content filtering—A way of controlling which content is delivered to certain users. ProxySG appliances can filter content based on content categories (such as gambling, games, and so on), type (such as http, ftp, streaming, and mime type), identity (user, group, network), or network conditions. You can filter content using vendor-based filtering or by allowing or denying access to URLs.

D

default boot system—The system that was successfully started last time. If a system fails to boot, the next most recent system that booted successfully becomes the default boot system.

default proxy listener—See *proxy service (default)*.

denial of service (DoS)—A method that hackers use to prevent or deny legitimate users access to a computer, such as a Web server. DoS attacks typically send many request packets to a targeted Internet server, flooding the server's resources and making the system unusable. Any system connected to the Internet and equipped with TCP-based network services is vulnerable to a DoS attack.

The ProxySG resists DoS attacks launched by many common DoS tools. With a hardened TCP/IP stack, the ProxySG resists common network attacks, including traffic flooding.

destination objects—Used in Visual Policy Manager. These are the objects that define the target location of an entry type.

detect protocol attribute—Detects the protocol being used. Protocols that can be detected include: HTTP, P2P (eDonkey, BitTorrent, FastTrack, Gnutella), SSL, and Endpoint Mapper.

diagnostic reporting—Found in the Statistics pane, the Diagnostics tab allows you to control whether Daily Heartbeats and/or Blue Coat Monitoring are enabled or disabled.

directives—Commands used in installable lists to configure forwarding and SOCKS gateway.

DNS access—A policy layer that determines how the ProxySG processes DNS requests.

domain name system (DNS)—An Internet service that translates domain names into IP addresses.

dynamic bypass—Provides a maintenance-free method for improving performance of the ProxySG by automatically compiling a list of requested URLs that return various kinds of errors.

dynamic real-time rating (DRTR)—Used in conjunction with the Blue Coat Web Filter (BCWF), DRTR (also known as *dynamic categorization*) provides real-time analysis and content categorization of requested Web pages to solve the problem of new and previously unknown uncategorized URLs—those not in the database.

When a user requests a URL that has not already been categorized by the BCWF database (for example, a brand new Web site), the ProxySG dynamic categorization service analyzes elements of the requested content and assigns a category or categories. The dynamic service is consulted *only* when the installed BCWF database does not contain category information for an object.

E

early intercept attribute—Controls whether the proxy responds to client TCP connection requests before connecting to the upstream server. When early intercept is disabled, the proxy delays responding to the client until after it has attempted to contact the server.

ELFF-compatible format—A log type defined by the W3C that is general enough to be used with any protocol.

emulated certificates—Certificates that are presented to the user by the ProxySG when intercepting HTTPS requests. Blue Coat emulates the certificate from the server and signs it, copying the subjectName and expiration. The original certificate is used between the ProxySG and the server.

encrypted log—A log is encrypted using an external certificate associated with a private key. Encrypted logs can only be decrypted by someone with access to the private key. The private key is not accessible to the ProxySG.

EULA—End user license agreement.

event logging—Allows you to specify the types of system events logged, the size of the event log, and to configure Syslog monitoring. The ProxySG can also notify you by email if an event is logged. *See also* access logging.

explicit proxy—A configuration in which the browser is explicitly configured to communicate with the proxy server for access to content. This is the default for the ProxySG and requires configuration for both the browser and the interface card.

extended log file format (ELFF)—A variant of the common log file format, which has two additional fields at the end of the line—the referer and the user agent fields.

F

fail open/closed—Failing open or closed applies to forwarding hosts and groups and SOCKS gateways. Fail open or closed applies when health checks are showing sick for each forwarding or SOCKS gateway target in the applicable fail-over sequence. If no systems are healthy, the ProxySG fails open or closed, depending on the configuration. If closed, the connection attempt simply fails.

If open, an attempt is made to connect without using any forwarding target (or SOCKS gateway). Fail open is usually a security risk; fail closed is the default if no setting is specified.

filtering—See *content filtering*.

forward proxy—A proxy server deployed close to the clients and used to access many servers. A forward proxy can be explicit or transparent.

FTP—See *Native FTP* and *Web FTP*.

G

gateway—A device that serves as entrance and exit into a communications network.

H

hardware serial number—A string that uniquely identifies the ProxySG; it is assigned to each unit in manufacturing.

health check tests—The method of determining network connectivity, target responsiveness, and basic functionality. The following tests are supported:

- ICMP
- TCP
- SSL
- HTTP
- HTTPS
- Group
- Composite and reference to a composite result
- ICAP
- Websense
- DRTR rating service

health check type—The kind of device or service the specific health check tests. The following types are supported:

- Forwarding host and forwarding group
- SOCKS gateway and SOCKS gateway group
- CAP service and ICAP service group
- Websense off-box service and Websense off-box service group
- DRTR rating service
- User-defined host and a user-defined composite

heartbeat—Messages sent once every 24 hours that contain the statistical and configuration data for the ProxySG, indicating its health. Heartbeats are commonly sent to system administrators and to Blue Coat. Heartbeats contain no private information, only aggregate statistics useful for pre-emptively diagnosing support issues.

The ProxySG sends emergency heartbeats whenever it is rebooted. Emergency heartbeats contain core dump and restart flags in addition to daily heartbeat information.

host affinity—The attempt to direct multiple connections by a single user to the same group member. Host affinity is closely tied to load balancing behavior; both should be configured if load balancing is important.

host affinity timeout—The host affinity timeout determines how long a user remains idle before the connection is closed. The timeout value checks the user's IP address, SSL ID, or cookie in the host affinity table.

I

inbound traffic (bandwidth gain)—Network packets flowing into the ProxySG. Inbound traffic mainly consists of the following:

- Server inbound: Packets originating at the origin content server (OCS) and sent to the ProxySG to load a Web object.

- Client inbound: Packets originating at the client and sent to the ProxySG for Web requests.

installable list—A list of configuration parameters that can be created using a text editor (either Blue Coat or another text editor) or through the CLI inline commands. The list can then be downloaded to the ProxySG from an HTTP server or locally from your PC. Configurations that can be created and installed this way include the SG Client, archiving, forwarding hosts, SOCKS gateways, ICP, policy files, and exceptions.

integrated host timeout—An integrated host is an origin content server (OCS) that has been added to the health check list. The host, added through the `integrate_new_hosts` property, ages out of the integrated host table after being idle for the specified time. The default is 60 minutes.

intervals—Time period from the completion of one health check to the start of the next health check.

IP reflection—Determines how the client IP address is presented to the origin server for explicitly proxied requests. All proxy services contain a reflect-ip attribute, which enables or disables sending of client's IP address instead of the IP address of the ProxySG.

issuer keyring—The keyring used by the ProxySG to sign emulated certificates. The keyring is configured on the appliance and managed through policy.

L

licensable component (LC)—(Software) A subcomponent of a license; it is an option that enables or disables a specific feature.

LCAMS—License Configuration and Management System.

license—Provides both the right and the ability to use certain software functions within a ProxyAV (or ProxySG) appliance. The license key defines and controls the license, which is owned by an account.

listener—The service that is listening on a specific port. A listener can be identified by any destination IP/subnet and port range. Multiple listeners can be added to each service.

live content—Also called live broadcast. Used in streaming, it indicates that the content is being delivered fresh.

LKF—License key file.

load balancing—A way to share traffic requests among multiple upstream systems or multiple IP addresses on a single host.

local bypass list—A list you create and maintain on your network. You can use a local bypass list alone or in conjunction with a central bypass list.

local policy file—Written by enterprises (as opposed to the central policy file written by Blue Coat); used to create company- and department-specific advanced policies written in the Blue Coat Policy Language (CPL).

log facility—A separate log that contains a single logical file and supports a single log format. It also contains the file's configuration and upload schedule information as well as other configurable information such as how often to rotate (switch to a new log) the logs at the destination, any passwords needed, and the point at which the facility can be uploaded.

log format—The type of log that is used: NCSA / Common, SQUID, ELFF, SurfControl, or Websense.

The proprietary log types each have a corresponding pre-defined log format that has been set up to produce exactly that type of log (these logs cannot be edited). In addition, a number of other ELFF type log formats are also pre-defined (im, main, p2p, ssl, streaming). These can be edited, but they start out with a useful set of log fields for logging particular protocols understood by the ProxySG. It is also possible to create new log formats of type ELFF or Custom which can contain any desired combination of log fields.

log tail—The access log tail shows the log entries as they get logged. With high traffic on the ProxySG, not all access log entries are necessarily displayed. However, you can view all access log information after uploading the log.

M

MACH5—SGOS 5 MACH5 Edition.

Management Console—A graphical Web interface that lets you to manage, configure, monitor, and upgrade the ProxySG from any location. The Management Console consists of a set of Web pages and Java applets stored on the ProxySG. The appliance acts as a Web server on the management port to serve these pages and applets.

management information base (MIB)—Defines the statistics that management systems can collect. A managed device (gateway) has one or more MIBs as well as one or more SNMP agents, which implements the information and management functionality defined by a specific MIB.

maximum object size—The maximum object size stored in the ProxySG. All objects retrieved that are greater than the maximum size are delivered to the client but are not stored in the ProxySG.

Media Access Control (MAC) address—A unique value associated with a network adapter; also known as hardware address or physical address. For the ProxySG, it is a hardware address that is stored in each network card (such as an SSL accelerator card or a Quad GigE Fiber LX card) on the ProxySG. The MAC address uniquely identifies an adapter on a LAN and is a 12-digit hexadecimal number (48 bits in length).

MIME/FILE type filtering—Allows organizations to implement Internet policies for both uploaded and downloaded content by MIME or FILE type.

multi-bit rate—The capability of a single stream to deliver multiple bit rates to clients requesting content from ProxySG appliances from within varying levels of network conditions (such as different connecting bandwidths and traffic).

multicast—Used in streaming; the ability for hundreds or thousands of users to play a single stream.

multicast aliases—Used in streaming; a streaming command that specifies an alias for a multicast URL to receive an .nsc file. The .nsc files allows the multicast session to obtain the information in the control channel

multicast station—Used in streaming; a defined location on the proxy where the Windows Media player can retrieve streams. A multicast station enables multicast transmission of Windows Media content from the cache. The source of the multicast-delivered content can be a unicast-live source, a multicast (live) source, and simulated live (video-on-demand content converted to scheduled live content).

multimedia content services—Used in streaming; multimedia support includes Real Networks, Microsoft Windows Media, Apple QuickTime, MP3, and Flash.

N

name inputing—Allows a ProxySG to resolve host names based on a partial name specification. When a host name is submitted to the DNS server, the DNS server resolves the name to an IP address. If the host name cannot be resolved, Blue Coat adds the first entry in the name-inputing list to the end of the host name and resubmits it to the DNS server

native FTP—Native FTP involves the client connecting (either explicitly or transparently) using the FTP protocol; the ProxySG then connects upstream through FTP (if necessary).

NCSA common log format—Blue Coat products are compatible with this log type, which contains only basic HTTP access information.

network address translation (NAT)—The process of translating private network (such as intranet) IP addresses to Internet IP addresses and vice versa. This methodology makes it possible to match private IP addresses to Internet IP addresses even when the number of private addresses outnumbers the pool of available Internet addresses.

non-cacheable objects—A number of objects are not cached by the ProxySG because they are considered non-cacheable. You can add or delete the kinds of objects that the appliance considers non-cacheable. Some of the non-cacheable request types are:

- Pragma no-cache, requests that specify non-cached objects, such as when you click refresh in the Web browser.
- Password provided, requests that include a client password.
- Data in request that include additional client data.
- Not a GET request.

.nsc file—Created from the multicast station definition and saved through the browser as a text file encoded in a Microsoft proprietary format. Without an .nsc file, the multicast station definition does not work.

NTP—To manage objects in an appliance, a ProxySG must know the current Universal Time Coordinates (UTC) time. By default, the ProxySG attempts to connect to a Network Time Protocol (NTP) server to acquire the UTC time. The ProxySG includes a list of NTP servers available on the Internet, and attempts to connect to them in the order they appear in the NTP server list on the NTP tab.

O

object (used in caching)—An object is the item that is stored in an appliance. These objects can be frequently accessed content, content that has been placed there by content publishers, or Web pages, among other things.

object (used in Visual Policy Manager)—An object (sometimes referred to as a condition) is any collection or combination of entry types you can create individually (user, group, IP address/subnet, and attribute). To be included in an object, an item must already be created as an individual entry.

object pipelining—This patented algorithm opens as many simultaneous TCP connections as the origin server will allow and retrieves objects in parallel. The objects are then delivered from the appliance straight to the user's desktop as fast as the browser can request them.

Online Certificate Status Protocol (OCSP)— An Internet protocol used for obtaining the revocation status of an X.509 digital certificate. OCSP was created as an alternative to certificate revocation lists (CRL), specifically addressing certain problems associated with using CRLs in a public key infrastructure (PKI). OCSP servers are called OCSP responders due to the request/response nature of these messages.

origin content server (OCS)—Also called origin server. This is the original source of the content that is being requested. An appliance needs the OCS to acquire data the first time, to check that the content being served is still fresh, and to authenticate users.

outbound traffic (bandwidth gain)—Network packets flowing out of the ProxySG. Outbound traffic mainly consists of the following:

- Client outbound: Packets sent to the client in response to a Web request.
- Server outbound: Packets sent to an OCS or upstream proxy to request a service.

P

PAC (Proxy AutoConfiguration) scripts—Originally created by Netscape, PACs are a way to avoid requiring proxy hosts and port numbers to be entered for every protocol. You need only enter the URL. A PAC can be created with the needed information and the local browser can be directed to the PAC for information about proxy hosts and port numbers.

packet capture (PCAP)—Allows filtering on various attributes of the Ethernet frame to limit the amount of data collected. You can capture packets of Ethernet frames going into or leaving a ProxySG.

parent class (bandwidth gain)—A class with at least one child. The parent class must share its bandwidth with its child classes in proportion to the minimum / maximum bandwidth values or priority levels.

passive mode data connections (PASV)—Data connections initiated by an FTP client to an FTP server.

pipelining—See *object pipelining*.

policies—Groups of rules that let you manage Web access specific to the needs of an enterprise. Policies enhance ProxySG feature areas such as authentication and virus scanning, and let you control end-user Web access in your existing infrastructure.

policy-based bypass list—Used in policy. Allows a bypass based on the properties of the client, unlike static and dynamic bypass lists, which allow traffic to bypass the appliance based on destination IP address. See also *dynamic bypass*.

policy layer—A collection of rules created using Blue Coat CPL or with the VPM.

pragma: no cache (PNC)—A metatag in the header of a request that requires the appliance to forward a request to the origin server. This allows clients to always obtain a fresh copy.

proxy—Caches content, filters traffic, monitors Internet and intranet resource usage, blocks specific Internet and intranet resources for individuals or groups, and enhances the quality of Internet or intranet user experiences.

A proxy can also serve as an intermediary between a Web client and a Web server and can require authentication to allow identity-based policy and logging for the client.

The rules used to authenticate a client are based on the policies you create on the ProxySG, which can reference an existing security infrastructure—LDAP, RADIUS, IWA, and the like.

Proxy Edition—SGOS 5 Proxy Edition.

proxy service—The proxy service defines the ports, as well as other attributes. that are used by the proxies associated with the service.

proxy service (default)—The default proxy service is a service that intercepts all traffic not otherwise intercepted by other listeners. It only has one listener whose action can be set to bypass or intercept. No new listeners can be added to the default proxy service, and the default listener and service cannot be deleted. Service attributes can be changed.

ProxySG—A Blue Coat security and cache box that can help manage security and content on a network.

public key certificate—An electronic document that encapsulates the public key of the certificate sender, identifies this sender, and aids the certificate receiver to verify the identity of the certificate sender. A certificate is often considered valid if it has been digitally signed by a well-known entity, which is called a Certificate Authority (such as VeriSign).

public virtual IP (VIP)—Maps multiple servers to one IP address and then propagates that information to the public DNS servers. Typically, there is a public VIP known to the public Internet that routes the packets internally to the private VIP. This enables you to “hide” your servers from the Internet.

R

real-time streaming protocol (RTSP)—A standard method of transferring audio and video and other time-based media over Internet-technology based networks. The protocol is used to stream clips to any RTP-based client.

reflect client IP attribute—Enables the sending of the client's IP address instead of the SG's IP address to the upstream server. If you are using an application delivery network (ADN), this setting is enforced on the concentrator proxy through the **Configuration > App. Delivery Network > Tunneling** tab.

registration—An event that binds the appliance to an account, that is, it creates the Serial#, Account association.

remote authentication dial-in user service (RADIUS)—Authenticates user identity via passwords for network access.

Return to Sender (RTS)—A way of allowing outgoing TCP packets to use the same network interface on which the corresponding incoming TCP packets arrived. The destination Media Acess Control (MAC) address for the outgoing packets is the same as the source MAC address of the incoming packets. See also *Media Access Control (MAC) address*.

reverse proxy—A proxy that acts as a front end to a small number of predefined servers, typically to improve performance. Many clients can use it to access the small number of predefined servers.

routing information protocol (RIP)—Designed to select the fastest route to a destination. RIP support is built into ProxySG appliances.

router hops—The number of jumps a packet takes when traversing the Internet.

RTS—See *Return to Sender.*

S

secure shell (SSH)—Also known as Secure Socket Shell. SSH is an interface and protocol that provides strong authentication and enables you to securely access a remote computer. Three utilities—login, ssh, and scp—comprise SSH. Security via SSH is accomplished using a digital certificate and password encryption. Remember that the Blue Coat ProxySG requires SSH1. A ProxySG supports a combined maximum of 16 Telnet and SSH sessions.

serial console—A third-party device that can be connected to one or more Blue Coat appliances. Once connected, you can access and configure the appliance through the serial console, even when you cannot access the appliance directly.

server certificate categories—The hostname in a server certificate can be categorized by BCWF or another content filtering vendor to fit into categories such as banking, finance, sports.

server portals—Doorways that provide controlled access to a Web server or a collection of Web servers. You can configure Blue Coat appliances to be server portals by mapping a set of external URLs onto a set of internal URLs.

server-side transparency—The ability for the server to see client IP addresses, which enables accurate client-access records to be kept. When server-side transparency is enabled, the appliance retains client IP addresses for all port 80 traffic to and from the ProxySG. In this scheme, the client IP address is always revealed to the server.

service attributes—Define the parameters, such as explicit or transparent, cipher suite, and certificate verification, that the ProxySG uses for a particular service.

sibling class (bandwidth gain)—A bandwidth class with the same parent class as another class.

signed system image—Cryptographically signed with a key known only to Blue Coat, and the signature is verified when the image is downloaded to the system.

simple network management protocol (SNMP)—The standard operations and maintenance protocol for the Internet. It consists of a set of standards for network management, including an application layer protocol, a database schema, and a set of data objects. In SNMP, the available information is defined by management information bases (MIBs), which describe the structure of the management data.

simulated live—Used in streaming. Defines playback of one or more video-on-demand files as a scheduled live event, which begins at a specified time. The content can be looped multiple times, or scheduled to start at multiple start times throughout the day.

SmartReporter log type—A proprietary ELFF log type that is compatible with the SmartFilter SmartReporter tool.

SOCKS—A proxy protocol for TCP/IP-based networking applications that allows users transparent access across the firewall. If you are using a SOCKS server for the primary or alternate forwarding gateway, you must specify the appliance's ID for the identification protocol used by the SOCKS gateway. The machine ID should be configured to be the same as the appliance's name.

SOCKS proxy—A generic way to proxy TCP and UDP protocols. The ProxySG supports both SOCKSv4/4a and SOCKSv5; however, because of increased username and password authentication capabilities and compression support, Blue Coat recommends that you use SOCKS v5.

splash page—The custom message page that displays the first time you start the client browser.

split proxy—Employs co-operative processing at the branch and the core to implement functionality that is not possible in a standalone proxy. Examples of split proxies include:

- Mapi Proxy
- SSL Proxy

SQUID-compatible format—A log type that was designed for cache statistics and is compatible with Blue Coat products.

squid-native log format—The Squid-compatible format contains one line for each request.

SSL authentication—Ensures that communication is with "trusted" sites only. Requires a certificate issued by a trusted third party (Certificate Authority).

SSL client—See SSL device profile.

SSL device profile—Used to determine various SSL parameters for outgoing HTTPS connections. Specifically, its role is to:

- Identify the SSL protocol version that the ProxySG uses in negotiations with origin servers.
- Identify the cipher suites used.
- Determine which certificate can be presented to origin servers by associating a keyring with the profile.

SSL interception—Decrypting SSL connections.

SSL proxy—A proxy that can be used for any SSL traffic (HTTPS or not), in either forward or reverse proxy mode.

static route—A manually-configured route that specifies the transmission path a packet must follow, based on the packet's destination address. A static route specifies a transmission path to another network.

statistics—Every Blue Coat appliance keeps statistics of the appliance hardware and the objects it stores. You can review the general summary, the volume, resources allocated, cache efficiency, cached contents, and custom URLs generated by the appliance for various kinds of logs. You can also check the event viewer for every event that occurred since the appliance booted.

stream—A flow of a single type of data, measured in kilobits per second (Kbps). A stream could be the sound track to a music video, for example.

SurfControl log type—A proprietary log type that is compatible with the SurfControl reporter tool. The SurfControl log format includes fully-qualified usernames when an NTLM realm provides authentication. The simple name is used for all other realm types.

syslog—An event-monitoring scheme that is especially popular in Unix environments. Most clients using Syslog have multiple devices sending messages to a single Syslog daemon. This allows viewing a single chronological event log of all of the devices assigned to the Syslog daemon. The Syslog format is: "Date Time Hostname Event."

system cache—The software cache on the appliance. When you clear the cache, all objects in the cache are set to expired. The objects are not immediately removed from memory or disk, but a subsequent request for any object requested is retrieved from the origin content server before it is served.

T

TCP window size—The number of bytes that can be buffered before the sending host must wait for an acknowledgement from the receiving host.

time-to-live (TTL) value—Used in any situation where an expiration time is needed. For example, you do not want authentication to last beyond the current session and also want a failed command to time out instead of hanging the box forever.

traffic flow (bandwidth gain)—Also referred to as *flow*. A set of packets belonging to the same TCP / UDP connection that terminate at, originate at, or flow through the ProxySG. A single request from a client involves two separate connections. One of

them is from the client to the ProxySG, and the other is from the ProxySG to the OCS. Within each of these connections, traffic flows in two directions—in one direction, packets flow out of the ProxySG (outbound traffic), and in the other direction, packets flow into the ProxySG (inbound traffic). Connections can come from the client or the server. Thus, traffic can be classified into one of four types:

- Server inbound
- Server outbound
- Client inbound
- Client outbound

These four traffic flows represent each of the four combinations described above. Each flow represents a single direction from a single connection.

transmission control protocol (TCP)—TCP, when used in conjunction with IP (Internet Protocol) enables users to send data, in the form of message units called packets, between computers over the Internet. TCP is responsible for tracking and handling, and reassembly of the packets; IP is responsible for packet delivery.

transparent proxy—A configuration in which traffic is redirected to the ProxySG without the knowledge of the client browser. No configuration is required on the browser, but network configuration, such as an L4 switch or a WCCP-compliant router, is required.

trial period—Starting with the first boot, the trial period provides 60 days of free operation. All features are enabled during this time.

U

unicast alias—Defines an name on the appliance for a streaming URL. When a client requests the alias content on the appliance, the appliance uses the URL specified in the unicast-alias command to request the content from the origin streaming server.

universal time coordinates (UTC)—A ProxySG must know the current UTC time. By default, the appliance attempts to connect to a Network Time Protocol (NTP) server to acquire the UTC time. If the ProxySG cannot access any NTP servers, you must manually set the UTC time.

URL filtering—*See* content filtering.

URL rewrite rules—Rewrite the URLs of client requests to acquire the streaming content using the new URL. For example, when a client tries to access content on www.mycompany.com, the ProxySG is actually receiving the content from the server on 10.253.123.123. The client is unaware that mycompany.com is not serving the content; however, the ProxySG access logs indicate the actual server that provides the content.

W

WCCP—Web Cache Communication Protocol. Allows you to establish redirection of the traffic that flows through routers.

Web FTP—Web FTP is used when a client connects in explicit mode using HTTP and accesses an ftp:// URL. The ProxySG translates the HTTP request into an FTP request for the OCS (if the content is not already cached), and then translates the FTP response with the file contents into an HTTP response for the client.

Websense log type—A Blue Coat proprietary log type that is compatible with the Websense reporter tool.

X

XML responder—HTTP XML service that runs on an external server.

XML requestor—XML realm.

Index

A

About 15
active client connections 146
ADN optimization
  attribute defined 30
Authenticate-401, attribute defined 29

B

bandwidth gain
  additional configurations affecting 135
  byte-range support 136
  revalidate pragma-no-cache effects 138
bandwidth refresh, configuring 135
browser
  proxy, configuring 214
  setting for explicit proxies 215
bypass list, overview 49
byte-range support
  bandwidth gain, affecting 136
  configuring 138

C

CCLs
  client certificates, specifying for 185
  server certificates, specifying for SSL proxy 185
client consent certificates, using with SSL proxy 186
creating an HTTP proxy service 114

D

destination IP address
  client
    trusted, configuring 44
    trusting 43
DNS
  destination IP address, trusting 43
DNS proxy
  overview 79
  resolving name list, explained 79
  resource record, creating 82
document, conventions 11
dynamic bypass
  configuring 51
  connection / receiving errors 51
  dynamic_timeout value 51
  lists, understanding 50
  max_dynamic_bypass_entry parameter 51
  server_bypass_threshold parameter 51
  troubleshooting 50
dynamic_timeout value, using with dynamic bypass 51

E

early intercept defined 29
editing an HTTP proxy service 114
explicit proxy
  browser settings 215
  creating 214
  Internet Explorer, using with 149
  overview 213
  ProxySG, using as proxy server 214
explicit TCP-Tunnel, explained 207

F

FTP
  clients, configuring 108
FTP proxy
  configuring 101

H

hardware models, licensing 45
HTTP object caching policy, configuring global defaults 123
HTTP object caching policy, customizing 119
HTTP object types 121
HTTP proxy
  about 113
  bandwidth gain profile 128
  bandwidth gain, fine-tuning 135
  byte-range support 136
  normal profile 128

portal profile 128
pragma-no-cache, revalidating 138
profile settings
configuring 133
explained 129
supporting IWA authentication for an explicit proxy 149
tolerant request parsing 122
HTTP proxy profile, configuring 133
HTTPS
origination 156
HTTPS console
creating 19
enabling 19
IP address, selecting 19
keyring, selecting 18
managing 19
HTTPS traffic, intercepting 181

I

intercepting all IPs on port 80 113
Internet Explorer, explicit proxy, using with 149
IP forwarding, enabling 217
issuer certificates, downloading for desktops 186
IWA
explicit proxy,Internet Explorer, using with 149
Internet Explorer, using with 149

L

license
hardware models, limits 45
user limits, managing 44

M

Management Console
managing 15
SSH
client keypairs, importing 23
configuring 21
Telnet console 24
max_dynamic_bypass_entry, using with dynamic bypass 51
meta tags, about 122
multiple listeners, best match 30

O

Object pipelining, about 120
objects, served 145
origination, HTTPS 156

P

PAC file, defined 214
Permeo
customer ID, obtaining 172
PA client, about 172
PA license, disabling on ProxySG 173
PA limitations 173
ProxySG, PA licensing 173
policy
bypass list 50
port services
attributes 29
HTTPS console, creating 19
supported 15
Telnet console, creating 24
prompt, customizing for Telnet 165
protocol detection 32
proxies
definition 11, 27
explicit, browser settings 215
explicit, creating 214
interface settings 215
SOCKS, configuring 170
proxies, understanding 213
proxy server, using ProxySG as 214
proxy services, best-match algorithm 30
proxy-support header
disabling through CPL 150
disabling through VPM 149
Internet Explorer, using with 149

R

realm banner, Telnet, customizing 165
reflect client IP address
client
reflect IP 43
refresh bandwidth, configuring 135
resolving name list, explained 79
restricted intercept
CLI, using 54
understanding 53

revalidate pragma-no-cache
bandwidth gain, using with 138
configuring 138
routing
bypass list 49
policy-based bypass list 50

S

server_bypass_threshold, dynamic bypass, using with 51
shell proxies
boundary conditions for 161
policy settings, customizing 160
*See also*Telnet
Telnet 161
understanding 159
SOCKS
compression gain statistics 175
connections, viewing 175
SOCKS clients, viewing 174
statistics 174
SOCKS proxy
bind timeout on accept value 171
configuring 170
connection timeout values 171
max-connection values 171
max-idle-timeout value 171
min-idle-timeout 171
SSH
client
keypairs , importing 23
managing 22
configuring 21
SSH host
welcome banner, creating 22
SSL proxy
Add Server Certificate object, configuring 191
Add SSL Forward Proxy object, configuring 190
categorizing hostnames in server certificates 192
CCLs for client certificates 185
CCLs, specifying for 185
client consent certificates, using 186
explicit mode, configuring 184
HTTPS
content, intercepting 190
traffic, intercepting 181
issuer certificates for desktops, downloading 186
rules, configuring 190
Server Certificate Category object, using 192
Set Server Certificate Validation object, using 193
SSL Access layer, using 192
SSL Intercept layer
configuring through CPL 194
using 190
statistics
unintercepted SSL byte 198
unintercepted SSL client 198
unintercepted SSL data 197
transparent mode, configuring 182
understanding 177
statistics
active client connections 146
client / server compression gain 147
HTTP / FTP bytes served 146
objects served 145
SOCKS clients, viewing 174
SSL proxy
unintercepted SSL bytes 198
unintercepted SSL clients 198
unintercepted SSL data 197

T

TCP-Tunnel
commands, explicit 211
explicit 207
overview 207
Telnet
banner settings, configuring 165
proxy boundary conditions 166
settings, customizing 165
shell proxy
understanding 161
Telnet console
error message 24
port service, explained 24
troubleshooting 24
tolerant request parsing, setting through CLI 122
transparent proxy

hardware, configuring 216
IP forwarding 217
IP forwarding, enabling 218
Layer-4 switch, using with 217
overview 213
troubleshooting
Telnet console 24
trust destination IP
configuring behavior 47

U

user license limits
behavior if exceeded 45
concurrent users, viewing 47
configuring behavior 47
license metrics, viewing 46
managing 44
notifications, setting 46

W

Web FTP
configuring IE 6.0 118
welcome banner, creating 22
welcome banner, Telnet, customizing for 165