Inktomi®

Traffic Edge Administrator's Guide

Release 1.5

June 2002

Content Networking Solutions Group
4100 East Third Avenue
Foster City, CA 94404

Phone: (650) 653-2800
Fax: (650) 653-2801
Web: http://www.inktomi.com

Contents

Preface

This manual describes how to use and configure an Inktomi Traffic Edge™ system.

For information about installing Traffic Edge, refer to the *Traffic Edge Installation Guide*. For information about unsupported features and last-minute information not available in this manual, refer to the *Release Notes*.

The manual discusses the following topics:

- *Chapter 1* provides an overview of Traffic Edge features and components.
- *Chapter 2* through *Chapter 15* provide procedural information about starting, monitoring, configuring, and maintaining Traffic Edge.
- *Appendix A* through *Appendix F* provide Traffic Edge reference information.
- *Appendix G* discusses frequently asked questions (FAQs) and provides troubleshooting tips.

Who Should Read This Manual

This manual is intended for Traffic Edge system administrators who configure, run, and administer Traffic Edge systems.

To use this manual, you should have working knowledge of web proxy caching, TCP/IP network protocols, network administration and management, and the UNIX or Windows operating system. If you have installed Traffic Edge Media Edition, you should be familiar with streaming media delivery.

Conventions Used in This Manual

This manual uses the following typographic conventions.

Convention	Purpose
italic	Represents emphasis and introduces terms: for example, “the *reverse proxy* option.”
bold	Represents graphical user interface options and menu names: for example, click the **Protocols** button.
`monospaced face`	Represents commands, filenames, file content, and computer input and output: for example, “use the `reconfigure` command.”
`monospaced italic`	Represents variables for which you should substitute a value: for example, “enter *`filename`*.”

brackets [ ]	Enclose optional command arguments in command syntax: for example, `add` *`pathname`* `[`*`size`*`]`.
vertical bar |	Separates value options in command syntax: for example, `open tcp|udp ports` *`o_ports`*.

CHAPTER 1

Overview

Traffic Edge speeds Internet access, enhances website performance, and delivers unprecedented web hosting capabilities.

This chapter discusses the following topics:

What Is Traffic Edge?

The dream of global data networking has come true. Internet users request billions of documents each day all over the world. Unfortunately, this dream of global data networking has become a nightmare for information systems professionals as they struggle with overloaded servers and congested networks, trying to keep pace with society's growing data demands.

Traffic Edge is a high-performance web proxy cache that improves network efficiency and performance by caching frequently accessed information at the *edge* of the network. This brings content physically closer to end users, for faster delivery, and dramatically reduces bandwidth use.

Traffic Edge is designed to improve content delivery for enterprises, Internet service providers (ISPs), backbone providers, and large intranets by maximizing existing bandwidth.

Traffic Edge Deployment Options

Traffic Edge can be deployed in different ways, to best suit your needs and your environment:

- As a web proxy cache
- As a reverse proxy
- In a cache hierarchy
- In a Traffic Edge cluster
- As a DNS proxy cache

The following sections provide a summary of these Traffic Edge deployment options.

Traffic Edge as a Web Proxy Cache

As a *web proxy cache*, Traffic Edge receives user requests for web content as those requests travel to the destined web server (origin server). If Traffic Edge contains the requested content, it serves the content directly; if not, Traffic Edge acts as a proxy, obtaining the content from the origin server on the user's behalf, while keeping a copy to satisfy future requests.

Traffic Edge provides two proxy caching options:

- *Transparent proxy caching*, in which user requests are automatically injected into a Traffic Edge cache on their way to the eventual destination. Users request Internet content as usual, without any browser configuration, and Traffic Edge automatically serves their requests. The user's client software (such as a browser or media player) is unaware that it is communicating with Traffic Edge. Transparent proxy caching is described in more detail in *Chapter 6, Transparent Proxy Caching*.
- *Explicit proxy caching*, in which the user's client software must be configured to send requests directly to Traffic Edge. Explicit proxy caching is described in more detail in *Chapter 5, Explicit Proxy Caching*.

Traffic Edge as a Reverse Proxy

As a *reverse proxy*, Traffic Edge is configured to be *the* origin server the user is trying to connect to (typically, the origin server's advertised hostname resolves to Traffic Edge, which is acting as the real origin server). The reverse proxy feature is also called *server acceleration*. Reverse proxy is described in more detail in *Chapter 7, Reverse Proxy and HTTP Redirects*.

Traffic Edge in a Cache Hierarchy

Traffic Edge can participate in flexible *cache hierarchies*, in which Internet requests not fulfilled in one cache can be routed to other regional caches, taking advantage of the contents and proximity of nearby caches. In a hierarchy of proxy servers, Traffic Edge can act either as a parent or a child cache, either to other Traffic Edge systems or to other caching products.

Traffic Edge supports parent caching and ICP (Internet Cache Protocol) peering. Hierarchical caching is described in more detail in *Chapter 9, Hierarchical Caching*.

Traffic Edge in a Cluster

Traffic Edge scales from a single node into multiple nodes that form a *cluster,* allowing you to improve system performance and reliability. Traffic Edge detects the addition or removal of nodes automatically. If the Traffic Edge *virtual IP failover* option is enabled, Traffic Edge maintains a pool of virtual IP addresses that it assigns to the nodes of the cluster. Traffic Edge can detect hard node failures (such as power supply or CPU failures) and reassign IP addresses of the failed node to the remaining operational nodes automatically.

Traffic Edge has two clustering modes:

- *Management only*, in which you can administer all the nodes in a cluster at the same time. Nodes automatically share configuration information.
- *Full clustering*, in which the node caches act as a single aggregate cache. A Traffic Edge cluster distributes its cache across its nodes into a single, virtual object store, rather than replicating the cache node by node.

  A fully clustered Traffic Edge system provides a single system image to both users and administrators, appearing as a single virtual server. Full-clustering mode includes management-only mode.

Traffic Edge clusters are described in more detail in *Chapter 8, Traffic Edge Clusters*.

Traffic Edge as a DNS Proxy Cache

As a DNS proxy cache, Traffic Edge can resolve DNS requests on behalf of clients. This option offloads remote DNS servers and reduces response time for DNS lookups; refer to *Chapter 11, DNS Proxy Caching*.

Traffic Edge Components

Traffic Edge consists of several components that work together to form a web proxy cache that you can easily monitor and configure. The main components are described below.

The Traffic Edge Cache

The *Traffic Edge cache* consists of a high-speed object database called the *object store*. The object store indexes objects according to URLs and associated headers. Using sophisticated object management, the object store can cache alternate versions of the same object, varying on spoken language or encoding type, and can efficiently store very small and very large objects, minimizing wasted space. When the cache is full, Traffic Edge removes stale data, ensuring that the most requested objects are kept on-hand and fresh.

Traffic Edge is designed to tolerate total disk failures on any of the cache disks. If the disk fails completely, Traffic Edge marks the entire disk as corrupt and continues using the remaining disks. If all of the cache disks fail, Traffic Edge goes into proxy-only mode.

You can partition the cache to reserve a certain amount of disk space for storing data for specific protocols and origin servers.

The Traffic Edge cache is described in more detail in *Chapter 10, Configuring the Cache*.

The RAM Cache

Traffic Edge maintains a small RAM cache of extremely popular objects. This *RAM cache* serves the most popular objects as fast as possible and reduces load on disks, especially during temporary traffic peaks. You can configure the RAM cache size to suit your needs; refer to *Changing the Size of the RAM Cache, on page 171*.

The Adaptive Redirection Module (ARM)

The Adaptive Redirection Module (ARM) is used in *transparent proxy caching* to redirect intercepted user requests destined for an origin server to Traffic Edge. Before the traffic is redirected by the ARM, it is intercepted by a Layer-4 switch or router.

To redirect user requests to Traffic Edge, the ARM changes an incoming packet's address. The packet's destination IP address is changed to the IP address of Traffic Edge, and the packet's destination port is changed according to the protocol used; for example, for HTTP, the packet's destination port is changed to the Traffic Edge HTTP port (usually 8080).

The ARM supports automatic bypass of sites that do not function properly with proxy caches.

The ARM also contains a connection-load shedding feature that prevents client request overloads. When there are more client connections than the specified limit, the ARM forwards incoming requests directly to the origin server; refer to *Connection Load Shedding (HTTP and FTP), on page 126*.

The Host Database

The Traffic Edge host database stores the domain name server (DNS) entries of origin servers to which Traffic Edge connects to fulfill user requests. This information is used to adapt future protocol interactions and optimize performance.

Among other information, the host database tracks:

- DNS information (for fast conversion of hostnames to IP addresses)
- The HTTP version of each host (so advanced protocol features can be used with hosts running modern servers)
- Host reliability and availability information (to avoid making the user wait for servers that are not running)

The DNS Resolver

Traffic Edge includes a fast, asynchronous DNS resolver to streamline conversion of hostnames to IP addresses. Traffic Edge implements the DNS resolver natively, directly issuing DNS command packets, rather than relying on slower, conventional resolver libraries. Many DNS queries can be issued in parallel and a fast DNS cache maintains popular bindings in memory, significantly reducing DNS traffic.

Traffic Edge Processes

Traffic Edge contains three processes that work together to serve Traffic Edge requests and manage, control, and monitor the health of the Traffic Edge system. The three processes are described below:

- The `traffic_server` process is the transaction processing engine of Traffic Edge. It is responsible for accepting connections, processing protocol requests, and serving documents from the cache or origin server.
- The `traffic_manager` process is the command and control facility of the Traffic Edge, responsible for launching, monitoring, and reconfiguring the `traffic_server` process. The `traffic_manager` process is also responsible for the Traffic Manager UI, the proxy autoconfiguration port, the statistics interface, cluster administration, and virtual IP failover.

  If the `traffic_manager` process detects a `traffic_server` process failure, it instantly restarts the process but also maintains a connection queue of all incoming requests. All incoming connections that arrive in the several seconds before full server restart are saved in the connection queue and processed in first-come, first-served order. This connection queueing shields users from any server restart downtime.
- The `traffic_cop` process monitors the health of both the `traffic_server` and `traffic_manager` processes. The `traffic_cop` process periodically (several times each minute) queries the `traffic_server` and `traffic_manager` process by issuing heartbeat requests to fetch synthetic web pages. In the event of failure (if no response is received within a timeout interval or if an incorrect response is received), `traffic_cop` restarts the `traffic_manager` and `traffic_server` processes.

Figure 1 illustrates the three Traffic Edge processes.

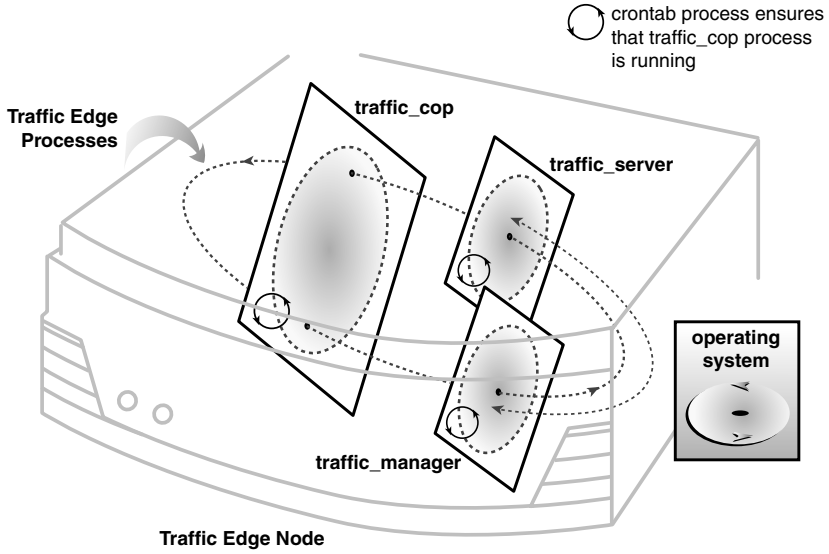


Figure 1 Traffic Edge processes

Administration Tools

Traffic Edge offers several administration alternatives to suit the needs of many environments:

- The *Traffic Manager* user interface (UI) is a web-based interface accessible through a browser. Traffic Manager provides a rich set of graphs and statistical displays for monitoring Traffic Edge performance and network traffic, and a set of options for configuring and fine-tuning the Traffic Edge system. Traffic Manager offers password-protected, SSL-encrypted, single-point administration for an entire Traffic Edge cluster.
- The *Traffic Line* command-line interface is a text-based interface, from which you can monitor Traffic Edge performance and network traffic and configure the Traffic Edge system. From Traffic Line, you can execute individual commands or script a series of commands in a shell.
- The *Traffic Shell* command-line interface is an additional command-line tool, from which you can monitor and configure the Traffic Edge system by executing individual commands.
- Various *configuration files* allow you to configure Traffic Edge through a simple file-editing and signal-handling interface. You can change configuration options by editing configuration files manually instead of using Traffic Manager, Traffic Line, or Traffic Shell. Any changes you make through Traffic Manager, Traffic Line, or Traffic Shell are automatically made to the configuration files.

Traffic Analysis Options

Traffic Edge provides several options for network traffic analysis and monitoring:

- *Traffic Manager statistics and graphs* show network traffic information. You can view graphs and statistics from Traffic Manager or collect and process statistics using Traffic Line or Traffic Shell.
- *MRTG* (Multi Router Traffic Grapher) is a graphing tool that provides a variety of graphs showing historical information about virtual memory usage, client connections, document hit rates, and so on. You can access MRTG from Traffic Manager.
- *SNMP Network Management* support lets you monitor and manage Traffic Edge through SNMP network management facilities. Traffic Edge supports two management information bases (MIBs): MIB-2, a well-known standard MIB, and the Inktomi proprietary Traffic Edge MIB that provides more specific node and cluster information.
- *Traffic Manager alarms* are presented in Traffic Manager. Traffic Edge signals an alarm for any detected failure condition. You can configure Traffic Edge to send email or page support personnel when an alarm occurs.
- *Transaction logging* lets you record information in a log file about every request that Traffic Edge receives and every error it detects. By analyzing the log files, you can determine how many people use the Traffic Edge cache, how much information each person requested, and what pages are most popular. You can also see why a particular transaction was in error and what state the Traffic Edge was in at a particular time; for example, you can see that Traffic Edge was restarted or that cluster communication timed out.

  Traffic Edge supports several standard log file formats, such as Squid and Netscape, and its own custom format. You can analyze the standard format log files with off-the-shelf analysis packages. To help with log file analysis, you can separate log files so that they contain information specific to protocol or hosts.

Traffic analysis options are described in more detail in *Chapter 12, Monitoring Traffic*. Traffic Edge logging options are described in *Chapter 15, Working with Log Files*.

Traffic Edge Security Options

Traffic Edge provides numerous options that enable you to establish secure communication between the Traffic Edge system and other computers on the network. Using the security options, you can do the following:

- Control client access to the Traffic Edge proxy cache.
- Control which hosts are allowed to access the Traffic Edge machine.
- Configure Traffic Edge integration into your firewall and control traffic through a SOCKS server.
- Configure Traffic Edge to use multiple DNS servers to match your site's security configuration; for example, Traffic Edge can use different DNS servers, depending on whether it needs to resolve hostnames located inside or outside a firewall. This enables you to keep your internal network configuration secure while continuing to provide transparent access to external sites on the Internet.
- Configure Traffic Edge to make sure that clients are authenticated before they can access content from the Traffic Edge cache. Traffic Edge supports LDAP, RADIUS, and NTLM proxy authentication and NTLM group authorization.
- Secure connections in reverse proxy mode between a client and Traffic Edge, and Traffic Edge and the origin server, using the SSL termination option.
- Control access to Traffic Manager using:
  - SSL (Secure Sockets Layer) protection for encrypted, authenticated access
  - An access control list (ACL) that defines which hosts are allowed to access Traffic Manager
  - User accounts that define which users can access Traffic Manager and which activities they can perform: for example, view statistics only or view statistics and configure Traffic Edge

Traffic Edge security options are described in more detail in *Chapter 14, Security Options*.

CHAPTER 2 Getting Started

After you have installed Traffic Edge on your system or on the nodes of your Traffic Edge cluster, you are ready to begin using Traffic Edge.

This chapter discusses the following topics:

Starting Traffic Edge

To start Traffic Edge, use the procedure appropriate for your operating system.

Starting Traffic Edge in UNIX

In UNIX, you can start Traffic Edge manually by issuing the `start_traffic_server` command. This command starts all the processes that work together to process Traffic Edge requests and manage, control, and monitor the health of the Traffic Edge system.

▼ **To run the start_traffic_server command:**

1. Log on to the Traffic Edge node as the Traffic Edge administrator and navigate to the Traffic Edge `bin` directory.
2. Enter the following command:

```
./start_traffic_server
```

Note Inktomi recommends that you always use the `start_traffic_server` command to start Traffic Edge.

Starting Traffic Edge in Windows

In Windows, you start Traffic Edge by running the Inktomi Traffic Cop service. By default, the Inktomi Traffic Cop service is set to automatic so that it starts whenever Windows boots. If the Inktomi Traffic Cop service is set to manual, you must start it manually from the **Services** control panel.

▼ **To start the Inktomi Traffic Cop service manually:**

1. Open the Control Panel, double-click the **Administrative Tools** icon and then double-click the **Services** icon.
2. Right-click the **Inktomi Traffic Cop** service and select **Start** from the context menu.

Verifying That Traffic Edge Is Running

After you have started Traffic Edge for the first time, verify that it is processing requests for web content.

▼ **To verify that Traffic Edge is processing requests:**

1. Access the Traffic Manager UI; refer to *Accessing Traffic Manager, on page 27*.
2. From the **Monitor** tab, click the **Protocols** button.
3. Click the **HTTP** button to display the **General HTTP Statistics** table (shown in *Figure 2*).
4. Make a note of the current **Total Document Bytes** statistic in the **Client** section of the table.

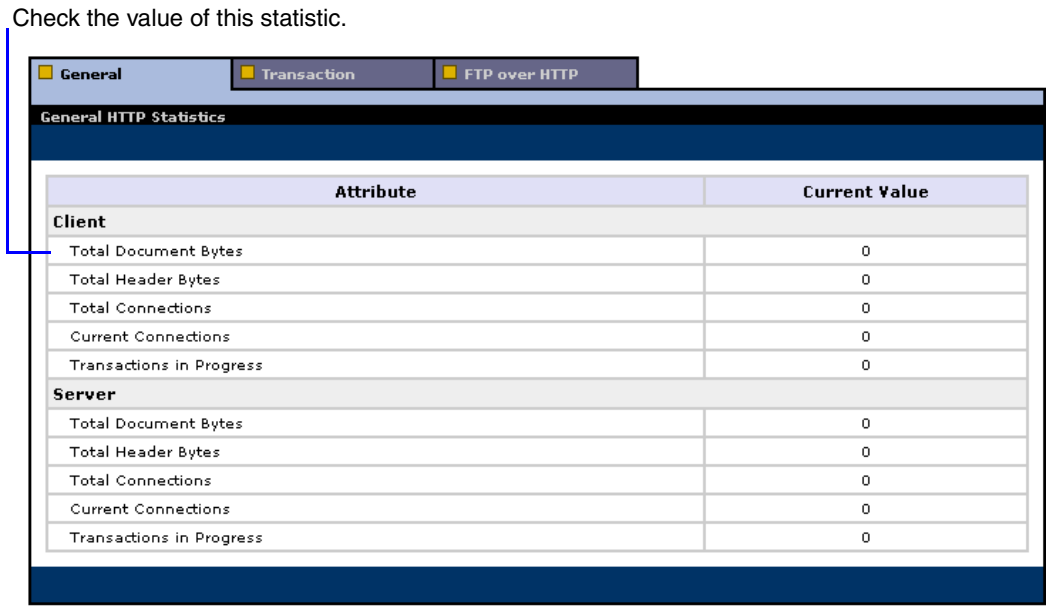


Figure 2 General HTTP statistics

5. Set your browser to the Traffic Edge proxy port.
6. Browse the Internet.
7. Recheck the **Total Document Bytes** statistic.

   This value increases as Traffic Edge processes HTTP requests.

Accessing Traffic Manager

Traffic Manager is the Traffic Edge browser-based user interface that provides a rich set of graphs and statistical displays for monitoring Traffic Edge performance and network traffic, and a set of options for configuring and fine-tuning your system.

You access Traffic Manager through your web browser.

▼ To access Traffic Manager:

1 Open your web browser.

Traffic Manager requires Java and JavaScript; be sure to enable Java and JavaScript in your browser.

2 Type one of the following locations in your browser:

Standard `http://nodename:adminport`

SSL `https://nodename:adminport`

nodename is the name of the Traffic Edge node and *adminport* is the number assigned to the Traffic Manager port (the default value for *adminport* is 8081).

Use the `https` address to reach Traffic Manager only if you have restricted access to Traffic Manager via SSL connections; otherwise, use the standard `http` address.

3 If necessary, log on to Traffic Edge with the administrator ID and password, or your user account. Traffic Manager opens in your web browser. *Figure 3, on page 28*, shows Traffic Manager.

The administrator ID and password are set during Traffic Edge installation. You can change the ID and password, as well as create and modify user accounts. For more information, refer to *Controlling Access to Traffic Manager, on page 206*.

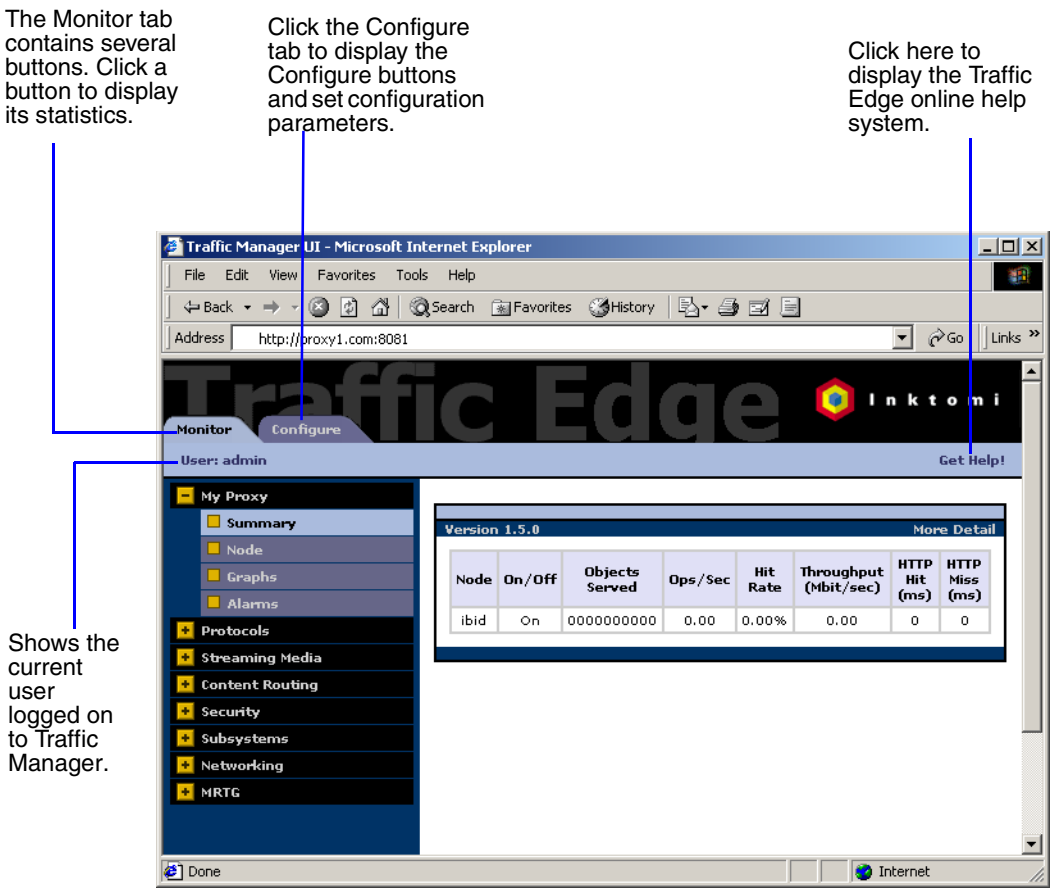


Figure 3 The Traffic Manager UI

Using the Monitor and Configure Tabs

Traffic Manager has two tabs:

- The **Monitor** tab lets you view Traffic Edge performance and network traffic statistics; refer to *Viewing Statistics from Traffic Manager, on page 180*.
- The **Configure** tab lets you view and modify Traffic Edge configuration options; refer to *Chapter 13, Configuring Traffic Edge*.

By default, Traffic Manager starts by displaying the **Monitor** tab.

Using Online Help

You can access the Traffic Edge online help system by clicking **Get Help** in the top right corner of the Traffic Manager display. Traffic Edge online help opens in separate browser window.

Starting Traffic Line

You can use Traffic Line to perform many of the tasks you can perform in Traffic Manager. Traffic Line provides a quick way of viewing Traffic Edge statistics and configuring the Traffic Edge system if you do not have access to a browser or if you prefer to use a UNIX shell-like command interface.

You can execute individual commands or script multiple commands in a shell; refer to *Appendix C, Traffic Line Commands*, for a list of commands.

▼ To start a Traffic Line session:

1. In UNIX, log on to a Traffic Edge node as the Traffic Edge administrator and navigate to the Traffic Edge `bin` directory. In Windows, open a Command Prompt window and then navigate to the Traffic Edge `bin` directory. This begins a Traffic Line session from which you can enter Traffic Line commands.

   Traffic Line commands take the following form:

   `traffic_line -command argument`

2. For a list of `traffic_line` commands, enter:

   `traffic_line -h`

   In UNIX, if the Traffic Edge `bin` directory is not in your path, prepend the Traffic Line command with `./` (for example, `./traffic_line -h`).

Starting Traffic Shell

Traffic Shell is a command-line tool that you can use to monitor and configure Traffic Edge. You can use Traffic Shell instead of using Traffic Manager or Traffic Line. Traffic Edge provides documentation for Traffic Shell in the form of manual pages (`man` pages). To start Traffic Shell and read an overview `man` page, use the following procedure.

Traffic Shell is supported on UNIX only.

▼ To start Traffic Shell:

1. Log on to a Traffic Edge node as the Traffic Edge administrator and navigate to the Traffic Edge `bin` directory.
2. Enter the following command:

   `./start_traffic_shell`

3. Enter the following command to display the `traffic_shell` overview man page:

   `man traffic_shell`

   The man page describes how to use Traffic Shell, how to obtain a list of available commands, and how to obtain documentation about each command.

Stopping Traffic Edge

To stop Traffic Edge, use the procedure appropriate for your operating system.

Stopping Traffic Edge in UNIX

In UNIX, you can stop Traffic Edge by issuing the `stop_traffic_server` command. The `stop_traffic_server` command stops all the Traffic Edge processes (`traffic_manager`, `traffic_server`, and `traffic_cop`).

Important Always use the `stop_traffic_server` command to stop Traffic Edge. Manually stopping processes can lead to unpredictable results.

▼ **To run the stop_traffic_server command:**

1. Log on to the node as the Traffic Edge administrator and navigate to the Traffic Edge `bin` directory.
2. Enter the following command:

   ```
   ./stop_traffic_server
   ```

Stopping Traffic Edge in Windows

In Windows, you stop Traffic Edge by stopping the Inktomi Traffic Cop service.

▼ **To stop the Inktomi Traffic Cop service manually:**

1. Open the Control Panel, double-click the **Administrative Tools** icon and then double-click the **Services** icon.
2. Right-click the **Inktomi Traffic Cop** service and select **Stop** from the context menu.

CHAPTER 3 HTTP and FTP Proxy Caching

Web proxy caching enables you to store copies of frequently accessed web objects (such as documents, images, and articles) close to users and serve this information to users on demand. Internet users get their information faster and Internet bandwidth is freed up for other tasks.

This chapter discusses the following topics:

Understanding HTTP Web Proxy Caching

Internet users direct their requests to web servers all over the Internet. For a caching server to serve these requests, it must act as a *web proxy server*. A web proxy server receives user requests for web objects and either serves the requests or forwards them to the *origin server* (the web server that contains the original copy of the requested information).

The Traffic Edge proxy supports both *transparent proxy caching*, in which the user's client software (typically a browser) is unaware that it is communicating with a proxy, and *explicit proxy caching*, in which the user's client software must be configured to send requests directly to the Traffic Edge proxy.

The following overview illustrates how Traffic Edge serves a user request.

Step 1 Traffic Edge receives a user request for a web object.

Step 2 Using the object address, Traffic Edge tries to locate the requested object in its object database (cache).

Step 3 If the object is in the cache, Traffic Edge checks to see if the object is fresh enough to serve. If so, Traffic Edge serves it to the user as a *cache hit* (*Figure 4*).

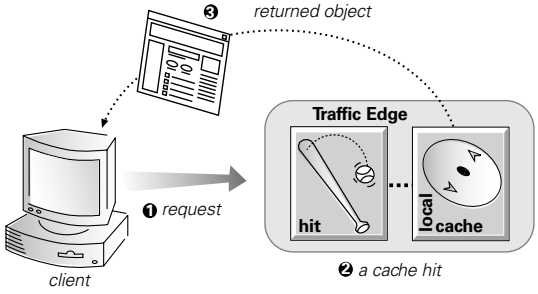


Figure 4 A cache hit

Step 4 If the data in the cache is stale, Traffic Edge connects to the origin server and checks if the object is still fresh (a revalidation). If so, Traffic Edge sends the cached copy to the user immediately.

Step 5 If the object is not in the cache (a *cache miss*) or the server indicates that the cached copy is no longer valid, Traffic Edge obtains the object from the origin server, simultaneously streaming it to the user and the cache (*Figure 5*). Subsequent requests for the object are served faster because the object will come directly from the cache.

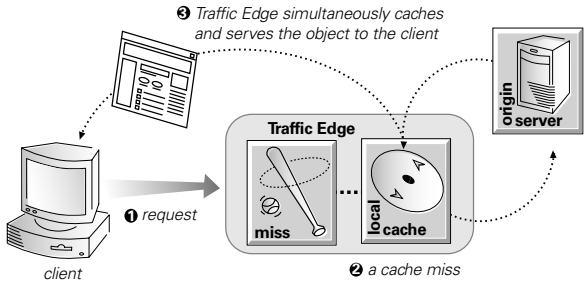


Figure 5 A cache miss

Caching is more complex than the preceding overview suggests. In particular, the overview does not discuss how Traffic Edge ensures freshness, serves correct HTTP alternates, and treats requests for objects that cannot or should not be cached. The following sections discuss these issues in detail.

Ensuring Cached Object Freshness

When Traffic Edge receives a request for a web object, it tries to locate the requested object in its cache. If the object is in the cache, Traffic Edge checks to see if the object is fresh enough to serve.

Traffic Edge handles object freshness in the cache differently, depending on protocol:

- HTTP objects support optional author-specified expiration dates. Traffic Edge adheres to these expiration dates; otherwise, it picks an expiration date based on how frequently the object is changing and on administrator-chosen freshness guidelines. In addition, objects can be revalidated, checking with the origin server if an object is still fresh; refer to *HTTP Object Freshness, on page 33*.
- FTP objects stay in the cache for an administrator-chosen time period; refer to *FTP Object Freshness, on page 39*.

HTTP Object Freshness

Traffic Edge determines whether an HTTP object in the cache is fresh by:

- Checking the `Expires` or `max-age` header

  Some HTTP objects contain `Expires` headers or `max-age` headers that explicitly define how long the object can be cached. Traffic Edge compares the current time with the expiration time to determine whether or not the object is fresh.

- Checking the `Last-Modified` / `Date` header

  If an HTTP object has no `Expires` header or `max-age` header, Traffic Edge can calculate a freshness limit using the following formula:

  `freshness_limit =(`*`date - last_modified`*`) * 0.10`

  `date` is the date in the object's server response header and *`last_modified`* is the date in the `Last-Modified` header. If there is no `Last-Modified` header, Traffic Edge uses the date that the object was written to cache. The value 0.10 (10 percent) can be increased or reduced to better suit your needs; refer to *Modifying the Aging Factor for Freshness Computations, on page 34*.

  The computed freshness limit is bound by a minimum and maximum freshness limit; refer to *Setting an Absolute Freshness Limit, on page 34*.

- Checking the absolute freshness limit

  For HTTP objects that do not have `Expires` headers or do not have both `Last-Modified` and `Date` headers, Traffic Edge uses a maximum and minimum freshness limit; refer to *Setting an Absolute Freshness Limit, on page 34*.

- Checking revalidate rules in the `cache.config` file

  Revalidate rules apply freshness limits to specific HTTP objects. You can set freshness limits for objects originating from particular domains or IP addresses, objects with URLs that contain specified regular expressions, objects requested by particular clients, and so on; refer to *cache.config, on page 369*.

Modifying the Aging Factor for Freshness Computations

If an object does not contain any expiration information, Traffic Edge can estimate its freshness from the `Last-Modified` and `Date` headers. By default, Traffic Edge stores an object for 10% of the time that elapsed since it last changed. You can increase or reduce the percentage to better suit your needs.

▼ **To modify the aging factor for freshness computations:**

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Edit the following variable:

Variable	Description
proxy.config.http.cache.heuristic_lm_factor	Set this variable to specify the aging factor for freshness computations. Traffic Edge stores an object for this percentage of the time that elapsed since it last changed. The default value is 0.10 (10 percent).

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

Setting an Absolute Freshness Limit

Some objects do not have `Expires` headers or do not have both `Last-Modified` and `Date` headers. You can control how long these objects are considered fresh in the cache by specifying an absolute freshness limit.

To specify an absolute freshness limit, you can either use Traffic Manager or edit a configuration file manually. Both procedures are provided below.

▼ **To specify an absolute freshness limit from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Protocols** button and then click the **HTTP** button.
3. Click the **Cacheability** tab.
4. Scroll down to the **Freshness** section.
5. In the **Minimum Heuristic Lifetime** field, specify the minimum amount of time that HTTP objects without an expiration date can remain fresh in the cache before being considered stale. The default value is 3600 seconds (1 hour).
6. In the **Maximum Heuristic Lifetime** field, specify the maximum amount of time that HTTP objects without an expiration date can remain fresh in the cache before being considered stale. The default value is 86400 seconds (1 day).
7. Click the **Apply** button.

▼ **To specify an absolute freshness limit manually:**

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Edit the following variables:

Variable	Description
proxy.config.http.cache.heuristic_min_lifetime	Set this variable to specify the minimum amount of time that HTTP objects without an expiration date can remain fresh in the cache before being considered stale. The default value is 3600 seconds (1 hour).
proxy.config.http.cache.heuristic_max_lifetime	Set this variable to specify the maximum amount of time that HTTP objects without an expiration date can remain fresh in the cache before being considered stale. The default value is 86400 seconds (1 day).

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

Specifying Header Requirements

To further ensure freshness of the objects in the cache, you can configure Traffic Edge to cache only objects with specific headers. You can do this either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

CAUTION By default, Traffic Edge caches all objects (including objects with no headers). Inktomi recommends that you change the default setting only for specialized proxy situations. If you configure Traffic Edge to cache only HTTP objects with `Expires` or `max-age` headers, the cache hit rate will be seriously reduced (very few objects have explicit expiration information).

▼ **To configure Traffic Edge to cache objects with specific headers from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Protocols** button and then click the **HTTP** button.
3. Click the **Cacheability** tab.
4. In the **Required Headers** area of the **Behavior** section, select one of the following options:
   - **An Explicit Lifetime Header** to cache only HTTP objects with `Expires` or `Cache-Control` headers.
   - **A Last-Modified Header** to cache only HTTP objects with `Expires` or `Last-Modified` headers.
   - **No Required Headers** to cache all HTTP objects (no specific headers are required). This is the default option.

5 Click the **Apply** button.

▼ **To configure Traffic Edge to cache objects with specific headers manually:**

1 In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.

2 Edit the following variable:

Variable	Description
proxy.config.http.cache.required_headers	Set this variable to one of the following values: 0 = no headers required for an HTTP object to be stored in the cache. 1 = at least last-modified header required for an HTTP object to be stored in the cache. 2 = Expires or max-age headers required for an HTTP object to be stored in the cache

3 Save and close the `records.config` file.

4 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

5 Run the command `traffic_line -x` to apply the configuration changes.

Cache-Control Headers

Even though an object might be fresh in the cache, clients or servers might have their own constraints that prevent them from retrieving the object from the cache; for example, a client might request that a object not come from a cache, or if it does, it cannot have been cached for more than 10 minutes.

Traffic Edge bases the servability of a cached object on `Cache-Control` headers. `Cache-Control` headers can appear in both client requests and server responses.

The following `Cache-Control` headers affect whether objects are served from the cache:

- The `no-cache` header, sent by clients, tells Traffic Edge to serve *no* objects directly from the cache; always obtain the object from the origin server. You can configure Traffic Edge to ignore client `no-cache` headers; refer to *Configuring Traffic Edge to Ignore Client no-cache Headers, on page 49*, below.
- The `max-age` header, sent by servers, is compared to the object age; if the age is less than `max-age`, the object is fresh and can be served.
- The `min-fresh` header, sent by clients, is an *acceptable freshness tolerance*. The client wants the object to be at least this fresh. If a cached object does not remain fresh at least this long in the future, it is revalidated.
- The `max-stale` header, sent by clients, permits Traffic Edge to serve stale objects provided they are not too old. Some browsers might be willing to take slightly old objects in exchange for improved performance, especially during periods of poor Internet availability.

Traffic Edge applies `Cache-Control` servability criteria *after* HTTP freshness criteria: for example, an object might be considered fresh, but if its age is greater than its `max-age`, it is not served.

Revalidating HTTP Objects

When a client requests an HTTP object that is stale in the cache, Traffic Edge *revalidates* the object. A revalidation is a query to the origin server to check if the object is unchanged. The result of a revalidation is one of the following:

- If the object is still fresh, Traffic Edge resets its freshness limit and serves the object.
- If a new copy of the object is available, Traffic Edge caches the new object, replacing the stale copy, and serves the object to the user simultaneously.
- If the object no longer exists on the origin server, Traffic Edge does not serve the cached copy.
- If the origin server does not respond to the revalidation query, Traffic Edge serves the stale object along with a `111 Revalidation Failed` warning.

By default, Traffic Edge revalidates a requested HTTP object in the cache if it considers the object to be stale. Traffic Edge evaluates object freshness as described in *HTTP Object Freshness, on page 33*.

You can reconfigure how Traffic Edge evaluates freshness by selecting one of the following options:

- Always revalidate HTTP objects in the cache with the origin server; Traffic Edge considers all HTTP objects in the cache to be stale.
- Never revalidate HTTP objects in the cache with the origin server; Traffic Edge considers all HTTP objects in the cache to be fresh.
- Revalidate all HTTP objects without `Expires` or `Cache-Control` headers; Traffic Edge considers all HTTP objects without `Expires` or `Cache-control` headers to be stale.

To configure how Traffic Edge revalidates objects in the cache, you can either use Traffic Manager or edit a configuration file manually. Both procedures are provided below.

In addition to the revalidation options listed above, you can also set specific revalidation rules in the `cache.config` file; refer to *cache.config, on page 369*.

▼ To configure revalidation options from Traffic Manager:

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Protocols** button and then click the **HTTP** button.
3. Click the **Cacheability** tab.
4. In the **When to Revalidate** area of the **Behavior** section, select:
   - **Never Revalidate** to never verify the freshness of a requested HTTP object with the origin server; Traffic Edge always considers HTTP objects to be fresh.
   - **Always Revalidate** to always verify the freshness of a requested HTTP object with the origin server; Traffic Edge always considers HTTP objects to be stale.

- **Revalidate if Heuristic Expiration** to verify the freshness of a requested HTTP object with the origin server if the object contains no `Expires` or `cache-control` headers; Traffic Edge considers all HTTP objects without `Expires` or `Cache-control` headers to be stale.
- **Use Cache Directive or Heuristic** to verify the freshness of a requested HTTP object with the origin server when Traffic Edge considers the object in the cache to be stale. This is the default option.

5 Click the **Apply** button.

▼ **To configure revalidation options manually:**

1 In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.

2 Edit the following variable:

proxy.config.http.cache.when_to_revalidate	Set this variable to one of the following options: 0 = configures Traffic Edge to revalidate an HTTP object whenever it is considered stale in the cache. (Traffic Edge checks the headers and the freshness limit, if applicable.) This is the default option. 1 = configures Traffic Edge to revalidate HTTP objects that do not contain Expires or cache-control headers. 2 = configures Traffic Edge to always revalidate HTTP objects; Traffic Edge always considers HTTP objects to be stale. 3 = configures Traffic Edge to never revalidate HTTP objects; Traffic Edge always considers HTTP objects to be fresh.

3 Save and close the `records.config` file.

4 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

5 Run the command `traffic_line -x` to apply the configuration changes.

FTP Object Freshness

FTP objects carry no timestamp or date information and remain fresh in the cache for the period of time you specify (from fifteen minutes to two weeks), after which they are considered stale.

FTP objects can be requested from either an HTTP client (such as a browser) or an FTP client (such as WS_FTP). Traffic Edge caches the FTP objects requested from HTTP clients in HTTP format and the FTP objects requested from FTP clients in a proprietary format.

The procedure for specifying the freshness limit for objects requested by HTTP clients is different from the procedure for objects requested by FTP clients. Follow the procedure appropriate for your needs.

FTP Objects Requested by HTTP Clients

You can set an absolute freshness limit for FTP objects requested by HTTP clients (FTP-over-HTTP objects) by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

In addition to setting an absolute freshness limit for all FTP-over-HTTP objects, you can set freshness rules for specific FTP-over-HTTP objects in the `cache.config` file; refer to *cache.config, on page 369*.

▼ To set the FTP-over-HTTP object freshness limit from Traffic Manager:

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Protocols** button and then click the **HTTP** button.
3. Click the **Cacheability** tab.
4. Scroll down to the **Freshness** section.
5. In the **FTP Document Lifetime** field, enter the amount of time that FTP objects requested by HTTP clients can remain fresh in the cache before being considered stale. The default value is 259200 seconds (3 days).
6. Click the **Apply** button.

▼ To set the FTP-over-HTTP object freshness limit manually:

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Edit the following variable:

Variable	Description
proxy.config.http.ftp.cache.document_lifetime	Set this variable to specify the number of seconds FTP objects requested from HTTP clients are considered fresh in the cache before being marked as stale. The default value is 259200 seconds (3 days).

3. Save and close the `records.config` file.

4 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

5 Run the command `traffic_line -x` to apply the configuration changes.

FTP Objects Requested by FTP Clients

You can set freshness limits for different types of FTP objects that are requested from FTP clients: for example, directory listings, login messages, and FTP files.

You set FTP freshness limits either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ To set FTP freshness limits from Traffic Manager:

1 From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.

2 On the **Configure** tab, click the **Protocols** button and then click the **FTP** button.

3 Click the **Cacheability** tab.

4 Scroll down to the **Freshness** section.

5 In the **Login Information** field, enter the amount of time that FTP login messages can remain fresh in the cache before Traffic Edge considers them stale. The default value is 2592000 seconds (30 days).

6 In the **Directory Listings** field, enter the amount of time that FTP directory listings can remain in the cache before Traffic Edge considers them stale. The default value is 604800 seconds (7 days).

7 In the **Files** field, enter the amount of time that FTP files can remain fresh in the cache before Traffic Edge considers them stale. The default value is 259200 seconds (3 days).

8 Click the **Apply** button.

▼ To set FTP freshness limits manually:

1 In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.

2 Edit the following variables:

Variable	Description
proxy.config.ftp.login_info_fresh_in_cache_time	Set this variable to specify the number of seconds that FTP login messages can remain fresh in the cache before Traffic Edge considers them stale. The default value is 2592000 seconds (30 days).

Variable	Description
proxy.config.ftp.directory_listing_fresh_in_cache_time	Set this variable to specify the number of seconds that FTP directory listings can remain in the cache before Traffic Edge considers them stale. The default value is 604800 seconds (7 days).
proxy.config.ftp.file_fresh_in_cache_time	Set this variable to specify the number of seconds that FTP files are considered fresh in the cache before Traffic Edge considers them stale. The default value is 259200 seconds (3 days).

3 Save and close the `records.config` file.

4 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

5 Run the command `traffic_line -x` to apply the configuration changes.

Scheduling Updates to Local Cache Content

To further increase performance and to ensure that HTTP and FTP-over-HTTP objects are fresh in the cache, you can use the scheduled update option to configure Traffic Edge to load specific objects into the cache at scheduled times. You might find this especially beneficial when using Traffic Edge as a reverse proxy so that you can preload content that you anticipate will be in demand.

Note The scheduled update option does not update FTP objects requested from FTP clients.

To use the scheduled update option, you must perform the following tasks.

- Specify the list of URLs that contain the objects you want to schedule for update, the time the update should take place, and the recursion depth for the URL
- Enable the scheduled update option and configure optional retry settings

Traffic Edge uses the information you specify to determine the URLs for which it is responsible and, for each URL, derives all recursive URLs if applicable. It then generates a unique URL list. Using this list, Traffic Edge initiates an HTTP `GET` for each unaccessed URL, ensuring that it remains within the user-defined limits for HTTP concurrency at any given time.

The system logs the completion of all HTTP `GET` operations, enabling you to monitor the performance of this feature.

Traffic Edge also provides a *Force Immediate Update* option that enables you to update URLs immediately without waiting for the specified update time to occur. You can use this option to test your scheduled update configuration; refer to *Forcing an Immediate Update, on page 43*.

Configuring the Scheduled Update Option

You can configure the scheduled update option either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ **To configure the scheduled update option from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Protocols** button and then click the **HTTP Scheduled Update** button.
3. Click the **Update URLs** tab.
4. In the **Scheduled Object Update** area, click the **Edit File** button.

   The configuration file editor for the `update.config` file opens.
5. In the fields provided, enter the following information:
   - In the **URL** field, enter the URL you want to schedule for update.

Optional

   - In the **Request Headers** field, enter the semicolon-separated list of headers passed in each `GET` request. You can define any request header that conforms to the HTTP specification.
   - In the **Offset Hour** field, enter the base hour used to derive the update periods. You can specify a value in the range 00 to 23.
   - In the **Interval** field, enter the interval (in seconds) at which updates occur, starting at the offset hour.
   - In the **Recursion Depth** field, enter the depth to which referenced URLs are recursively updated, starting at the given URL; for example, a recursion depth of 1 will update the given URL, as well as all URLs immediately referenced by links from the original URL.
6. Click the **Add** button and then click the **Apply** button.
7. Click the **Close** button to exit the configuration file editor.
8. Click the **General** tab.
9. Enable the **Scheduled Update** option.
10. In the **Maximum Concurrent Updates** field, enter the maximum number of simultaneous update requests allowed at any point in time. This option prevents the scheduled update process from overburdening the host. The default value is 100.
11. In the **Count** field of the **Retry on Update Error** section, enter the number of times you want to retry the scheduled update of a URL in the event of failure. The default value is 10.
12. In the **Interval** field of the **Retry on Update Error** section, enter the delay in seconds between each scheduled update retry for a URL in the event of failure. The default value is 2.
13. Click the **Apply** button.

▼ **To configure the scheduled update option manually:**

1. In a text editor, open the `update.config` file located in the Traffic Edge `config` directory.
2. Enter a line in the file for each URL you want to update; refer to *update.config, on page 456*.
3. Save and close the `update.config` file.
4. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
5. Edit the following variables:

Variable	Description
proxy.config.update.enabled	Set this variable to 1 to enable the scheduled update option.
proxy.config.update.retry_count	Set this variable to specify the number of times you want to retry the scheduled update of a URL in the event of failure. The default value is 10.
proxy.config.update.retry_interval	Set this variable to specify the delay in seconds between each scheduled update retry for a URL in the event of failure. The default value is 2.
proxy.config.update.concurrent_updates	Set this variable to specify the maximum simultaneous update requests allowed at any point in time. This option enables you to prevent the scheduled update process from overburdening the host. The default value is 100.

6. Save and close the `records.config` file.
7. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
8. Run the command `traffic_line -x` to apply the configuration changes.

Forcing an Immediate Update

Traffic Edge provides a Force Immediate Update option that lets you verify the URLs listed in the `update.config` file immediately. The Force Immediate Update option disregards the offset hour and interval set in the `update.config` file and immediately updates the URLs listed.

You can enable the Force Immediate Update option either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ **To enable the Force Update option from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Protocols** button and then click the **HTTP Scheduled Update** button.

3 On the **General** tab, make sure that the **Scheduled Update** option is enabled.

4 Click the **Update URLs** tab.

5 Enable the **Force Immediate Update** option.

6 Click the **Apply** button.

▼ **To enable the Force Update Option manually:**

1 In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.

2 Edit the following variable:

Variable	Description
proxy.config.update.force	Set this variable to 1 to enable the Force Immediate Update option.

3 Make sure that the variable `proxy.config.update.enabled` is set to 1.

4 Save and close the `records.config` file.

5 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

6 Run the command `traffic_line -x` to apply the configuration changes.

IMPORTANT When you enable the Force Immediate Update option, Traffic Edge continually updates the URLs specified in the `update.config` file until you disable the option. To disable the Force Immediate Update option, set the variable `proxy.config.update.force` to 0 (zero).

Pushing Content into the Cache

Traffic Edge supports the HTTP PUSH method of content delivery. Using HTTP PUSH, you can deliver content directly into the cache without user request.

Configuring Traffic Edge to Accept PUSH Requests

Before you can deliver content into your cache using HTTP PUSH, you must configure Traffic Edge to accept PUSH requests. You can either use Traffic Manager or edit configuration files manually. Both procedures are provided below.

▼ **To configure Traffic Edge to accept PUSH requests from Traffic Manager:**

1 From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.

2 On the **Configure** tab, click the **Protocols** button and then click the **HTTP** button.

3 On the **General** tab, enable the **PUSH Method** option and then click the **Apply** button.

4 On the **Configure** tab, click the **Security** button and then click the **Access Control** button.

The configuration file editor for the `filter.config` file opens.

5 Add a rule for each IP address or IP address range allowed to deliver PUSH requests, as follows:

- From the **Rule Type** drop-down box, select **allow**.
- From the **Primary Destination Type** drop-down box, select **dest_domain**.
- In the **Primary Destination Value** field enter . (a period).
- In the **Source IP** field in the **Secondary Specifiers** area, enter the IP address or IP address range allowed to deliver PUSH requests.
- From the **Method** drop-down list box, select **PUSH**.

Note **PUSH** displays in the drop-down list box only when **PUSH Method** is enabled in the **General** tab under **Configure/Protocols/HTTP**.

- Click the **Add** button to add the rule to the `filter.config` file.

6 Add a rule to deny all other hosts the ability to deliver PUSH requests:

- From the **Rule Type** drop-down box, select **deny**.
- From the **Primary Destination Type** drop-down box, select **dest_domain**.
- In the **Primary Destination Value** field enter . (a period).
- From the **Method** drop-down list box, select **PUSH**.
- Click the **Add** button to add the rule to the `filter.config` file.

7 Click the **Apply** button to save the rules you created.

8 Click the **Close** button to exit the configuration file editor.

▼ To configure Traffic Edge to accept PUSH requests manually:

1 In a text editor, open the `filter.config` file located in the Traffic Edge `config` directory.

2 Add the following filter rules to the file to ensure that only certain IP addresses can deliver PUSH requests to the cache:

```
domain=. src_ip=ipaddress method=PUSH action=allow
```

```
domain=. method=PUSH action=deny
```

ipaddress is the IP address of the host or range of IP addresses of the hosts from which Traffic Edge accepts PUSH requests.

3 Save and close the `filter.config` file.

4 In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.

5 Edit the following variable:

Variable	Description
proxy.config.http.push_method_enabled	Set this variable to 1 to enable Traffic Edge to accept PUSH requests.

6 Save and close the `records.config` file.

7 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

8 Run the command `traffic_line -x` to apply the configuration changes.

Understanding HTTP PUSH

PUSH uses the HTTP 1.1 message format. The body of a PUSH request contains the response header and response body that you want to place in the cache. The following is an example of a PUSH request:

```
PUSH http://www.company.com HTTP/1.0
Content-length: 84

HTTP/1.0 200 OK
Content-type: text/html
Content-length: 17

<HTML>
a
</HTML>
```

IMPORTANT Your header must include content-length. Content-length must include both header and body byte count.

Pinning Content in the Cache

The cache pinning option configures Traffic Edge to keep certain HTTP and FTP-over-HTTP objects in the cache for a specified time. You can use this option to ensure that the most popular objects are in the cache when needed and that Traffic Edge does not delete important objects.

Traffic Edge observes Cache-Control headers and pins an object in the cache only if it is cacheable.

To use the cache pinning option, you must perform the following tasks:

- Set cache pinning rules in the `cache.config` file.
- Enable the cache pinning option.

You can perform these tasks either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ To set cache pinning rules and enable cache pinning from Traffic Manager:

1 From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.

2 On the **Configure** tab, click the **Protocols** button and then click the **HTTP** button.

3 Click the **Cacheability** tab.

4 Scroll to the **Caching Policy/Forcing Document Caching** section at the end of the page.

5 Click the **Edit File** button.

The configuration file editor for the `cache.config` file opens.

6 In the fields provided, supply the following information:

- From the **Rule Type** drop-down box, select **pin-in-cache**.
- From the **Primary Destination Type** drop-down box, select **url_regex**.
- In the **Primary Destination Value** field, specify the URL you want to pin in the cache.
- In the **Time Period** field, specify the amount of time that Traffic Edge pins the object in the cache.

In addition, you can add secondary specifiers (such as **Prefix** and **Suffix**) to the rule. All the fields are described under *HTTP, on page 306*.

7 Click the **Add** button to add the rule to the list and then click the **Apply** button to save the rule.

8 Click the **Close** button to exit the configuration file editor.

9 On the **Configure** tab, click the **Subsystems** button and then click the **Cache** button.

10 Enable the **Allow Pinning** option on the **General** tab.

11 Click the **Apply** button.

▼ To set cache pinning rules and enable cache pinning manually:

1 In a text editor, open the `cache.config` file located in the Traffic Edge `config` directory.

2 Add a rule in the file for each URL you want Traffic Edge to pin in the cache, as shown below.

`url_regex=`*`URL`* `pin-in-cache=12h`

`URL` is the URL you want Traffic Edge to pin in the cache. The time format can be `d` for days, `h` for hours (as shown), `m` for minutes, and `s` for seconds. You can also use mixed units: for example, `1h15m20s`. In addition, you can add secondary specifiers (such as prefix and suffix) to the rule; refer to *cache.config, on page 369* for more information.

3 Save and close the `cache.config` file.

4 In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.

5 Edit the following variable:

Variable	Description
proxy.config.cache.permit.pinning	Set this variable to 1 to enable the cache pinning option.

6 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

7 Run the command `traffic_line -x` to apply the configuration changes.

To Cache or Not to Cache?

When Traffic Edge receives a request for a web object that is not in the cache, it retrieves the object from the origin server and serves it to the client. At the same time, Traffic Edge checks if the object is cacheable before storing it in its cache to serve future requests.

Traffic Edge determines if an object is cacheable differently, depending on protocol:

- For HTTP objects, Traffic Edge responds to caching directives from clients and origin servers. In addition, you can configure Traffic Edge not to cache certain objects; refer to *Caching HTTP Objects, on page 48*.
- For FTP objects, Traffic Edge responds to caching directives you specify through configuration options and files; refer to *Caching FTP Objects, on page 59*.

Caching HTTP Objects

Traffic Edge responds to caching directives from clients and origin servers, as well as directives you specify through configuration options and files.

Client Directives

By default, Traffic Edge does *not* cache objects with the following request headers:

- `Cache-Control: no-store` header
- `Cache-Control: no-cache` header

  You can configure Traffic Edge to ignore the `Cache-Control: no-cache` header; refer to *Configuring Traffic Edge to Ignore Client no-cache Headers*, below.
- `Cookie:` header (for text objects)

  By default, Traffic Edge caches objects served in response to requests that contain cookies unless the object is text. You can configure Traffic Edge to *not* cache cookied content of any type, cache all cookied content, or cache cookied content that is of image type only; refer to *Caching Cookied Objects, on page 53*.
- `Authorization:` header

FTP objects requested from HTTP clients can also contain `Cache-Control: no-store`, `Cache-Control: no-cache`, or `Authorization` headers. If an FTP object requested from an HTTP client contains such a header, Traffic Edge does not cache it unless explicitly configured to do so.

Configuring Traffic Edge to Ignore Client no-cache Headers

By default, Traffic Edge strictly observes client `Cache Control:no-cache` directives. If a requested object contains a `no-cache` header, Traffic Edge forwards the request to the origin server even if it has a fresh copy in the cache.

You can configure Traffic Edge to ignore client `no-cache` directives. In this case, Traffic Edge ignores `no-cache` headers from client requests and serves the object from its cache.

To configure Traffic Edge to ignore client `no-cache` headers, you can either use Traffic Manager or edit a configuration file manually. Both procedures are provided below.

IMPORTANT The default behavior of observing `no-cache` directives is appropriate in most cases. Only configure Traffic Edge to ignore client `no-cache` directives if you are knowledgeable about HTTP 1.1.

▼ To configure Traffic Edge to ignore client no-cache headers from Traffic Manager:

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Protocols** button and then click the **HTTP** button.
3. Click the **Cacheability** tab.
4. In the **Behavior** section, enable the **Ignore "no-cache" in Client Requests** option.
5. Click the **Apply** button.

▼ To configure Traffic Edge to ignore client no-cache headers manually:

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Edit the following variable:

Variable	Description
proxy.config.http.cache.ignore_client_no_cache	Set this variable to 1 to ignore client requests to bypass the cache.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

Note Certain versions of Microsoft Internet Explorer do not request cache reloads from reverse proxies and transparent caches when the user presses the browser **Refresh** button. This can prevent content from being loaded directly from the origin server. You can configure Traffic Edge to treat Microsoft Internet Explorer requests more conservatively, providing fresher content at the cost of serving fewer documents from the cache. You can configure Traffic Edge to add no-cache headers to requests from Microsoft Internet Explorer in Traffic Manager (in the **Behavior** section of the **Cacheability** tab under **Configure/Protocols/HTTP**). Alternatively, you can set the configuration variable `proxy.config.http.cache.when_to_add_no_cache_to_msie_requests` in the `records.config` file to 1 (add `no-cache` to `If Modified Since` MSIE requests) or 2 (add `no-cache` to all MSIE requests).

Origin Server Directives

By default, Traffic Edge does not cache objects with the following response headers:

- `Cache-Control: no-store` header
- `Cache-Control: private` header
- `WWW-Authenticate`: header

  You can configure Traffic Edge to ignore `WWW-Authenticate` headers; refer to *Configuring Traffic Edge to Ignore WWW-Authenticate Headers, on page 51*.
- `Set-Cookie`: header
- `Cache-Control: no-cache` headers

  You can configure Traffic Edge to ignore `no-cache` headers; refer to *Configuring Traffic Edge to Ignore Server no-cache Headers, on page 50*.
- `Expires`: header with value of 0 (zero) or a past date

Configuring Traffic Edge to Ignore Server no-cache Headers

By default, Traffic Edge strictly observes `Cache-Control:no-cache` directives. A response from an origin server with a `no-cache` header is not stored in the cache and any previous copy of the object in the cache is removed.

You can configure Traffic Edge to ignore origin server `no-cache` headers.

IMPORTANT If you configure Traffic Edge to ignore `no-cache` headers, Traffic Edge also ignores `no-store` headers.

IMPORTANT The default behavior of observing `no-cache` directives is appropriate in most cases. Only configure Traffic Edge to ignore origin server `no-cache` headers if you are knowledgeable about HTTP 1.1.

▼ **To configure Traffic Edge to ignore server no-cache headers:**

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Edit the following variable:

Variable	Description
proxy.config.http.cache.ignore_server_no_cache	Set this variable to 1 to ignore server directives to bypass the cache.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

Configuring Traffic Edge to Ignore WWW-Authenticate Headers

By default, Traffic Edge does not cache objects that contain `WWW-Authenticate` response headers. The `WWW-Authenticate` header contains authentication parameters that the client uses when preparing the authentication challenge response to an origin server.

You can configure Traffic Edge to ignore origin server `WWW-Authenticate` headers, in which case, objects with `WWW-Authenticate` headers are stored in the cache for future requests.

IMPORTANT The default behavior of not caching objects with `WWW-Authenticate` headers is appropriate in most cases. Only configure Traffic Edge to ignore server `WWW-Authenticate` headers if you are knowledgeable about HTTP 1.1.

▼ **To configure Traffic Edge to ignore server WWW-Authenticate headers:**

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Edit the following variable:

Variable	Description
proxy.config.http.cache.ignore_authentication	Set this variable to 1 to cache objects with WWW Authenticate headers.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

Configuration Directives

In addition to client and origin server directives, Traffic Edge responds to directives you specify through configuration options and files.

You can configure Traffic Edge to:

- *Not* cache any HTTP objects; refer to *Disabling HTTP Object Caching, on page 52*.
- Cache dynamic content (objects with URLs that contain a question mark (?), a semicolon (;), or cgi or that end in `.asp`); refer to *Caching Dynamic Content, on page 52*.
- Cache objects served in response to the `Cookie`: header; refer to *Caching Cookied Objects, on page 53*.
- Observe never-cache rules in the `cache.config` file; refer to *cache.config, on page 369*.

Disabling HTTP Object Caching

By default, Traffic Edge caches all HTTP objects except those for which you have set never-cache rules in the `cache.config` file. You can disable HTTP object caching so that all HTTP objects are served directly from the origin server and never cached.

▼ **To disable HTTP object caching from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Protocols** button and then click the **HTTP** button.
3. Click the **Cacheability** tab.
4. Disable the **HTTP Caching** option.
5. Click the **Apply** button.

▼ **To disable HTTP object caching manually:**

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Edit the following variable:

Variable	Description
proxy.config.http.cache.http	Set this variable to 0 (zero) to disable HTTP object caching.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

Caching Dynamic Content

A URL is considered dynamic if it contains a question mark (?), a semicolon (;), or cgi or if it ends in `.asp`. By default, Traffic Edge does *not* cache dynamic content. However, you can configure Traffic Edge to cache dynamic content either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

CAUTION Inktomi recommends that you configure Traffic Edge to cache dynamic content for specialized proxy situations only.

▼ **To configure Traffic Edge to cache dynamic content:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Protocols** button and then click the **HTTP** button.
3. Click the **Cacheability** tab.
4. In the **Dynamic Caching** section, enable the **Caching Documents with Dynamic URLs** option.
5. Click the **Apply** button.

▼ **To configure Traffic Edge to cache dynamic content manually:**

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Edit the following variable:

Variable	Description
proxy.config.http.cache_urls_that_look_dynamic	Set this variable to 1 to cache dynamic content.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

Caching Cookied Objects

By default, Traffic Edge caches objects served in response to requests that contain cookies *unless* the object is text. Traffic Edge does not cache cookied text content because object headers are stored as well as the object, and personalized cookie header values could be saved with the object. With nontext objects, it is unlikely that personalized headers are delivered or used.

You can reconfigure Traffic Edge to:

- Not cache cookied content of any type
- Cache cookied content that is of image type only
- Cache all cookied content regardless of type

You can configure how Traffic Edge caches cookied content either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below:

▼ **To configure how Traffic Edge caches cookied content from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Protocols** button and then click the **HTTP** button.
3. Click the **Cacheability** tab.
4. In the **Caching Response to Cookies** area of the **Dynamic Caching** section, select the caching option you want to use:
   - Select **Cache All but Text** to cache all cookied content except content that is text (this is the default setting).
   - Select **Cache Only Image Types** to cache cookied content that is an image.
   - Select **Cache Any Content Type** to cache cookied content of all types.
   - Select **No Cache on Cookies** to *not* cache cookied content of any type.
5. Click the **Apply** button.

▼ **To configure how Traffic Edge caches cookied content manually:**

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Edit the following variable:

Variable	Description
proxy.config.http.cache.cache_responses_to_cookies	Set this variable to specify how Traffic Edge caches cookied content: 0 = Do not cache any responses to cookies. 1 = Cache all responses to cookies. 2 = Cache responses to cookies of image type only. 3 = Cache all responses to cookies except text content-types (the default).

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

Forcing Object Caching

You can force Traffic Edge to cache specific URLs (including dynamic URLs) for a specified duration regardless of Cache-Control response headers.

You can specify the URLs you want to force and the duration either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ **To force object caching from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Protocols** button and then click the **HTTP** button.
3. Click the **Cacheability** tab.
4. Scroll to the **Caching Policy/Force Document Caching** section at the end of the page.
5. Click the **Edit File** button.

   The configuration file editor for the `cache.config` file opens.
6. In the fields provided, supply the following information:
   - From the **Rule Type** drop-down box, select **ttl-in-cache**.
   - From the **Primary Destination Type** drop-down box, select **url_regex**.

- In the **Primary Destination Value** field, specify the URL you want to force cache.
- In the **Time Period** field, specify the amount of time that Traffic Edge can serve the URL from the cache.

In addition, you can add secondary specifiers (such as **Prefix** and **Suffix**) to the rule. All the fields are described in *HTTP, on page 306*.

7 Click the **Add** button to add the rule to the list and then click the **Apply** button to save the rule.

8 Click the **Close** button to exit the configuration file editor.

▼ **To force document caching manually:**

1 In a text editor, open the `cache.config` file located in the Traffic Edge `config` directory.

2 Add a rule in the file for each URL you want Traffic Edge to force cache, as shown below.

```
url_regex=URL ttl-in-cache=6h
```

URL is the URL you want Traffic Edge to force cache. The time format can be `d` for days, `h` for hours (as shown), `m` for minutes, and `s` for seconds. You can also use mixed units: for example, `1h15m20s`. In addition, you can add secondary specifiers (for example, prefix and suffix) to the rule; refer to *cache.config, on page 369*.

3 Save and close the `cache.config` file.

4 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

5 Run the command `traffic_line -x` to apply the configuration changes.

Caching HTTP Alternates

Some origin servers answer requests to the same URL with a variety of objects. The content of these objects can vary widely, according to whether a server delivers content for different languages, targets different browsers with different presentation styles, or provides different document formats (HTML, PDF). Different versions of the same object are termed *alternates* and are cached by Traffic Edge based on `Vary` response headers.

You can specify additional request and response headers for specific content types that Traffic Edge will identify as alternates for caching. You can also limit the number of alternate versions of an object allowed in the cache.

Configuring How Traffic Edge Caches Alternates

You can either use Traffic Manager or edit a configuration file manually to specify additional request and response headers that Traffic Edge will identify as alternates for caching. Both procedures are provided below.

▼ To configure how Traffic Edge caches alternates from Traffic Manager:

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Protocols** button and then click the **HTTP** button.
3. Click the **Cacheability** tab.
4. Scroll down to the **Vary Based on Content Type** section.
5. Click **Enabled** to cache alternate versions of HTTP documents that do not contain the `Vary` header.
6. Specify the additional request and response headers you want Traffic Edge to identify:
   - In the **Vary by Default on Text** field, enter the HTTP header field on which you want to vary if the request is for text: for example, an HTML document.
   - In the **Vary by Default on Images** field, enter the HTTP header field on which you want to vary if the request is for images: for example, a `.gif` file.
   - In the **Vary by Default on Other Document Types** field, enter the HTTP header field on which you want to vary if the request is for anything other than text or images.

Note If you specify **Cookie** as the header field on which to vary in the above fields, make sure that the appropriate option is enabled in the **Caching Response to Cookies** area of the **Dynamic Caching** section. For example, if you enable the **Cache Only Image Types** option in the **Caching Response to Cookies** area, and you enable the **Vary by Default on Text** option in the **Vary Based on Content Type** section, alternates by cookie will not apply to text.

7. Click the **Apply** button.

▼ To configure how Traffic Edge caches alternates manually:

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Edit the following variables:

Variable	Description
proxy.config.http.cache.enable_default_vary_headers	Set this variable to 1 to cache alternate versions of HTTP objects that do not contain the Vary header.
proxy.config.http.cache.vary_default_text	Set this variable to specify the HTTP header field on which you want to vary if the request is for text: for example, an HTML document.
proxy.config.http.cache.vary_default_images	Set this variable to specify the HTTP header field on which you want to vary if the request is for images: for example, a `.gif` file.
proxy.config.http.cache.vary_default_other	Set this variable to specify the HTTP header field on which you want to vary if the request is for anything other than text or image.

Note If you specify Cookie as the header field on which to vary in the above variables, make sure that the `proxy.config.http.cache.cache_responses_to_cookies` variable is set appropriately. For example, if you set the `proxy.config.http.cache.cache_responses_to_cookies` variable to 2 (cache responses to cookies of image type only) and set the `proxy.config.http.cache.vary_default_text` variable to specify cookie, alternates by cookie will not apply to text.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

Limiting the Number of Alternates for an Object

You can limit the number of alternates Traffic Edge can cache per object. The default number of alternates is three. You can either use Traffic Manager or edit a configuration file manually to set the maximum number of alternates allowed in the cache.

IMPORTANT Large numbers of alternates can affect Traffic Edge cache performance because all alternates have the same URL. Although Traffic Edge can look up the URL in the index very quickly, it must scan sequentially through available alternates in the object store.

▼ To limit the number of alternates from Traffic Manager:

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Protocols** button and then click the **HTTP** button.
3. Click the **Cacheability** tab.
4. In the **Maximum Alternates** field, enter the maximum number of alternate versions of an object you want Traffic Edge to cache. The default value is three.
5. Click the **Apply** button.

▼ To limit the number of alternates manually:

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Edit the following variable:

Variable	Description
proxy.config.cache.limits.http.max_alts	Set this variable to specify the maximum number of alternate versions of an object you want Traffic Edge to cache. The default value is three.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

Using Congestion Control

The congestion control option lets you configure Traffic Edge to stop forwarding HTTP requests to origin servers when they become congested and to send the client a message to retry the congested origin server later.

To use the congestion control option, you must perform the following tasks:

- Enable the congestion control option.
- Create rules in the `congestion.config` file to specify:
  - Which origin servers Traffic Edge tracks for congestion
  - The timeouts Traffic Edge uses depending on whether a server is congested
  - The page that Traffic Edge sends to the client when a server becomes congested
  - If Traffic Edge tracks the origin servers per IP address or per hostname
  - If Traffic Edge sends SNMP traps when a server becomes congested

You can enable and configure the congestion control option either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ To enable and configure the congestion control option from Traffic Manager:

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Networking** button and then click the **Connection Management** button.
3. Click the **Congestion Control** tab.
4. Enable the **Congestion Control** option.
5. The **Congestion Rules** section displays the `congestion.config` file. Scroll to the end of the file and enter rules that specify which servers are tracked for congestion and the timeout values Traffic Edge uses to determine congestion; refer to *congestion.config, on page 371* for the rule format.
6. Click the **Apply** button to save your configuration changes.

▼ To enable and configure the congestion control option manually:

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Edit the following variable:

Variable	Description
proxy.config.http.congestion_control.enabled	Set this variable to 1 to enable the congestion control option.

3. Save and close the `records.config` file.
4. In a text editor, open the `congestion.config` file located in the Traffic Edge `config` directory.

5 Enter rules that specify which origin servers are tracked for congestion and the timeout values Traffic Edge uses to determine congestion; refer to *congestion.config, on page 371* for the rule format.

6 Save and close the `congestion.config` file.

7 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

8 Run the command `traffic_line -x` to apply the configuration changes.

Caching FTP Objects

FTP objects can be requested from either an HTTP client (such as a browser) or an FTP client (such as WS_FTP). Traffic Edge treats these objects differently.

For FTP objects requested from HTTP clients (FTP-over-HTTP), you can perform the following configuration to determine what Traffic Edge stores in the cache:

- Disable FTP-over-HTTP caching so that Traffic Edge does not cache any FTP objects requested from HTTP clients; refer to *Disabling FTP-Over-HTTP Caching*, below.
- Set never cache rules in the `cache.config` file; refer to *cache.config, on page 369*.
- Configure Traffic Edge to ignore client `Cache-Control: no-store` or `Cache-Control: no-cache` headers; refer to *Configuring Traffic Edge to Ignore Client no-cache Headers, on page 49*.

For FTP objects requested from FTP clients (FTP proxy objects), you can perform the following configuration to determine what Traffic Edge stores in the cache:

- Disable FTP proxy object caching so that Traffic Edge does not cache any FTP objects requested from FTP clients; refer to *Disabling FTP Proxy Object Caching, on page 60*.
- Disable caching of simple and/or full directory listings; refer to *Disabling Caching of Full or Simple Directory Listings, on page 61*.

Disabling FTP-Over-HTTP Caching

You can configure Traffic Edge not to cache any FTP objects that are requested from HTTP clients by disabling the FTP-over-HTTP option. Traffic Edge will still process the requests by forwarding them directly to the FTP server but will not cache any requested objects.

You can disable the FTP-over-HTTP caching option either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ To disable FTP-over-HTTP caching from Traffic Manager:

1 From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.

2 On the **Configure** tab, click the **Protocols** button and then click the **HTTP** button.

3 Click the **Cacheability** tab.

4 In the **Caching** section, disable the **FTP-over-HTTP Caching** option.

5 Click the **Apply** button.

▼ **To disable FTP-over-HTTP caching manually:**

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Edit the following variable:

Variable	Description
proxy.config.http.cache.ftp	Set this variable to 0 (zero) to disable FTP-over-HTTP caching.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

Disabling FTP Proxy Object Caching

You can configure Traffic Edge not to cache any FTP objects that are requested from FTP clients. Traffic Edge will still process FTP requests by forwarding them directly to the FTP server (as long as the **FTP** processing option is enabled in the **Features** table on the **My Proxy/Basic** tab) but will not cache any requested objects.

You can disable FTP proxy object caching either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ **To disable FTP proxy object caching from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Protocols** button and then click the **FTP** button.
3. Click the **Cacheability** tab.
4. Disable the **FTP Caching** option.
5. Click the **Apply** button.

▼ **To disable FTP proxy object caching manually:**

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Edit the following variable:

Variable	Description
proxy.config.ftp.cache_enabled	Set this variable to 0 (zero) to disable FTP object caching for requests that come from FTP clients. When you disable this variable, Traffic Edge still processes FTP objects but does not cache them (Traffic Edge forwards all requests for FTP objects to the FTP server). If you want Traffic Edge to stop processing FTP requests, you must set the variable proxy.config.ftp.ftp_enabled to 0.

3 Save and close the `records.config` file.

4 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

5 Run the command `traffic_line -x` to apply the configuration changes.

Disabling Caching of Full or Simple Directory Listings

You can configure Traffic Edge not to cache simple and/or full directory listings. A simple directory listing contains no arguments: for example, `dr/ls`. A full directory listing does contain arguments: for example, `ls *.txt`.

▼ **To disable caching of simple and/or full directory listings from Traffic Manager:**

1 From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.

2 On the **Configure** tab, click the **Protocols** button and then click the **FTP** button.

3 Click the **Cacheability** tab.

4 In the **Directory Caching** section:

- Disable the **Simple** option if you do not want to cache simple directory listings.
- Disable the **Full** option if you do not want to cache full directory listings.

5 Click the **Apply** button.

▼ **To disable caching of simple and/or full directory listings manually:**

1 In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.

2 Edit the following variables:

Variable	Description
proxy.config.ftp.simple_directory_listing_cache_enabled	Set this variable to 0 (zero) if you do *not* want to cache simple directory listings.
proxy.config.ftp.full_directory_listing_cache_enabled	Set this variable to 0 (zero) if you do *not* want to cache full directory listings.

3 Save and close the `records.config` file.

4 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

5 Run the command `traffic_line -x` to apply the configuration changes.

CHAPTER 4 Streaming Media Proxy Caching

Traffic Edge supports proxy caching of streaming media content.

Streaming media content (*streams)* can be audio clips or video clips. Streams can be either live (in real time) or on-demand (previously recorded).

This chapter discusses the following topics:

Understanding Streaming Media Proxy Caching

Traffic Edge can process and cache RTSP requests from QuickTime media systems, RTSP and PNA requests from Real Networks media systems, and MMS (Microsoft Media Streaming) requests from Windows media systems.

Traffic Edge can send out the same live stream to multiple clients (*live splitting*) and can broadcast live streams from an Inktomi Media Distribution Network (refer to the *Inktomi Media Distribution Network Installation and Configuration Guide*).

The following section describes Traffic Edge support for QuickTime, Real Networks, and Windows media systems. For information about the streaming media server and player software versions that are compatible with Traffic Edge, refer to the *Traffic Edge Installation Guide*.

Using QuickTime

When you enable the QuickTime option (either during or after installation), Traffic Edge serves and caches QuickTime streams.

Traffic Edge supports the following deployment options for Quicktime:

- *Explicit proxy caching*; refer to *Explicit Proxy Caching for QuickTime Requests, on page 90*.
- *Transparent proxy caching*; refer to *Transparency for QuickTime Requests, on page 101*.
- *Reverse proxy caching*; refer to *Reverse Proxy for QuickTime Requests, on page 143*.
- *Hierarchical caching*; refer to *Chapter 9, Hierarchical Caching*.
- *Application- and network-level firewall* and *SOCKS* support; refer to *Firewall Support for Streaming Media, on page 235* and *Configuring SOCKS Firewall Integration, on page 210*.

For information about configuring general streaming media options, such as enabling streaming media proxy caching, changing the RTSP proxy port, configuring memory-based throttling, and configuring streaming using HTTP, refer to *Configuring General Streaming Media Options, on page 65*. For information about configuring specific QuickTime options, such as MediaBridge monitoring, refer to *Configuring QuickTime Options, on page 69*.

Using Real Networks

When you enable the Real Networks option (either during or after installation), Traffic Edge serves and caches Real Networks streams.

Traffic Edge works together with the RealProxy software from Real Networks to provide proxy caching for Real Networks media systems. To support Real Networks, you must install a licensed copy of RealProxy on your Traffic Edge system; refer to the *Traffic Edge Installation Guide*.

Traffic Edge supports the following deployment options for Real Networks:

- *Explicit proxy caching*; refer to *Explicit Proxy Caching for Real Media Player Requests, on page 92*.
- *Transparent proxy caching*; refer to *Transparency for Real Media Player Requests, on page 102*.
- *Reverse proxy caching*; refer to *Reverse Proxy for Real Media Player Requests, on page 144*.
- *Network-level firewall support*; refer to *Configuring Network-Level Firewalls for Real Networks, on page 235*.

For information about configuring general streaming media options, such as enabling streaming media proxy caching, changing the RTSP proxy port, and configuring streaming using HTTP, refer to *Configuring General Streaming Media Options, on page 65*. For information about configuring specific Real Networks options, such as changing the default port for explicit requests, configuring RealProxy tunneling, setting the RealProxy restart limit, configuring passthrough, and using multicast, refer to *Configuring RealProxy Options, on page 70*.

Using WMT

When you enable the WMT option (either during or after installation), Traffic Edge serves and caches WMT streams.

WMT uses metafiles to allow users to play WMT streams on web browsers. When a user clicks a link to a metafile in a browser, the metafile starts the Windows Media Player, which requests the desired stream. The metafile contains the URL for the stream, which includes the name of a file in a WMT media streaming format, usually with an `.asf` or `.wmv` extension. The Windows Media Player typically uses the MMS protocol to send requests to the Windows Media Server.

Traffic Edge intercepts web browser requests for WMT streams and rewrites the media content URL in the metafile to point to Traffic Edge instead of the Windows Media Server.

Traffic Edge can process and cache both MMS and HTTP requests for WMT streams.

Traffic Edge supports the following deployment options for WMT:

- *Explicit proxy caching*; refer to *Explicit Proxy Caching for WMT Requests, on page 93*.
- *Transparent proxy caching*; refer to *Transparency for WMT Requests, on page 103*.
- *Reverse proxy caching*; refer to *Reverse Proxy for WMT Requests, on page 145*.
- *Hierarchical caching*; refer to *Chapter 9, Hierarchical Caching*.
- *Application- and network-level firewall* and *SOCKS* support; refer to *Firewall Support for Streaming Media, on page 235* and *Configuring SOCKS Firewall Integration, on page 210*.

For information about configuring general streaming media options, such as enabling streaming media proxy caching, configuring memory-based throttling, and configuring streaming using HTTP, refer to *Configuring General Streaming Media Options, on page 65*. For information about configuring specific WMT options, such as changing the MMS proxy port, disabling ASX file rewrite, configuring memory for WMT retransmissions, using media push and multicast, and configuring MediaBridge monitoring, refer to *Configuring WMT Options, on page 74*.

Configuring General Streaming Media Options

This section discusses the following topics:

Enabling Streaming Media Proxy Caching

To process and cache streaming media requests, you must enable the streaming media options. If you select the streaming media options during installation, the installation script performs the configuration automatically; otherwise, you can either use Traffic Manager or edit a configuration file manually. Both procedures are provided below.

Note To support Real Networks, you must install a licensed copy of RealProxy on your Traffic Edge system; refer to the *Traffic Edge Installation Guide*.

▼ **To configure Traffic Edge to process streaming media requests from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. Click the **Configure** tab.

   The **General** tab of the **My Proxy**/**Basic** button displays.
3. In the **Streaming Media** section of the **Features** table, select the type of streaming media requests you want to process (QuickTime, WMT, and/or Real Networks).
4. Click the **Apply** button.

▼ **To configure Traffic Edge to process streaming media requests manually:**

1. In a text editor, open the `records.config` file located in the `config` directory.
2. Edit the following variables:

Variable	Description
proxy.config.qt.enabled	Set this variable to 1 to enable Traffic Edge to process QuickTime requests.
proxy.config.rni.enabled	Set this variable to 1 to enable Traffic Edge to process Real media player requests.
proxy.config.wmt.enabled	Set this variable to 1 to enable Traffic Edge to process WMT requests.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -L` to restart Traffic Edge on the local node or `traffic_line -M` to restart Traffic Edge on all the nodes in a cluster.

Changing the RTSP Proxy Port

Traffic Edge uses the RTSP proxy port to accept all QuickTime requests and all transparent and reverse proxy Real media player requests. By default, the RTSP proxy port is set to 554. You can change the port either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

To change the proxy port for explicit Real media player requests, refer to *Changing the Default Port for Explicit Requests, on page 71*.

To change the MMS proxy port, refer to *Changing the MMS Proxy Port, on page 74*.

▼ **To change the RTSP proxy port from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Streaming Media** button and then click the **Shared Settings** button.
3. In the **RTSP Proxy Port** field, enter the port number you want to use for all QuickTime requests and all transparent and reverse proxy Real media player requests.
4. Click the **Apply** button.

▼ **To change the RTSP proxy port manually:**

1. In a text editor, open the `records.config` file located in the `config` directory.
2. Edit the following variable:

Variable	Description
proxy.config.mixt.rtsp_proxy_port	Set this variable to specify the port number you want Traffic Edge to use for all QuickTime requests and all transparent and reverse proxy Real media player requests.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

Configuring Memory-Based Throttling

You can configure Traffic Edge to deny all new QuickTime and WMT client connections when a certain amount of memory is in use. You can set this value by editing a configuration file manually.

Note This setting does not apply to Real media player connections.

▼ **To configure memory-based throttling:**

1. In a text editor, open the `records.config` file located in the `config` directory.
2. Edit the following variable:

Variable	Description
proxy.config.resource.target_maxmem_mb	Set this variable to specify the maximum amount of memory in use before Traffic Edge starts to deny new QuickTime and WMT connections. You can enter a value in the range 0 - 65536. A value of 0 (zero) means that Traffic Edge ignores this setting (no connections are denied).

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

Configuring Streaming Using HTTP

Streaming using the HTTP protocol is typically used by the web browser after preferred strategies have failed and is not commonly used as the primary streaming strategy for a Traffic Edge streaming media deployment.

Streaming using HTTP makes one separate connection from the client to the Traffic Edge node and another separate connection from the Traffic Edge node to the origin server. Traffic Edge automatically terminates HTTP connections when the default HTTP timeout values are reached.

If your deployment uses HTTP as the streaming protocol, you must modify the HTTP timeout values so that Traffic Edge does not terminate HTTP transactions before the streaming is complete. You must determine which streaming content has the longest duration, add some time to that duration, and convert the total to seconds.

You can modify HTTP timeouts either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ To modify HTTP Timeouts from Traffic Manager:

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Protocols** button and then click the **HTTP** button.
3. Click the **Timeouts** tab.
4. Set the inactivity and activity timeouts.

   Inktomi recommends that you set all the timeouts to a large value: for example, 1452.
5. Click the **Apply** button.

▼ To modify HTTP timeouts manually:

1. In a text editor, open the `records.config` file located in the `config` directory.
2. Edit the following variables.

Note Inktomi recommends that you set all the timeouts to a large value: for example, 14520.

Variable	Description
proxy.config.http.transaction_no_activity_timeout_in	Set this variable to specify how long Traffic Edge must keep connections to clients open if a transaction stalls. The default value is 120 seconds.
proxy.config.http.transaction_no_activity_timeout_out	Set this variable to specify how long Traffic Edge must keep connections to origin servers open if the transaction stalls. The default value is 120 seconds.

Variable	Description
proxy.config.http.transaction_active_timeout_in	Set this variable to specify the maximum amount of time Traffic Edge can remain connected to a client. If the transfer to the client is not complete before this timeout expires, Traffic Edge closes the connection. The default value of 0 specifies that there is no timeout.
proxy.config.http.transaction_active_timeout_out	Set this variable to specify the maximum amount of time Traffic Edge waits for fulfillment of a connection request to an origin server. If Traffic Edge does not complete the transfer to the origin server before this timeout expires, Traffic Edge terminates the connection request. The default value of 0 specifies that there is no timeout.

3 Save and close the `records.config` file.

4 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

5 Run the command `traffic_line -x` to apply the configuration changes.

Configuring QuickTime Options

If your Traffic Edge node is part of an Inktomi Media Distribution Network, you can configure MediaBridge monitoring for QuickTime either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below. For information about the Inktomi Media Distribution Network, refer to the Inktomi Media Distribution Network documentation.

▼ **To configure MediaBridge monitoring for QuickTime from Traffic Manager:**

1 From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.

2 On the **Configure** tab, click the **Streaming Media** button and then click the **QuickTime** button.

3 In the **Media Bridge Name** field, enter the name of the MediaBridge node.

4 In the **Media Bridge Port** field, enter the port number of the MediaBridge node. The default port is 10036.

5 In the **Media Bridge Mount Point** field, enter the mount point for MDN streams.

6 In the **Monitor Name** field, enter the name of the monitoring agent.

7 In the **Monitor Port** field, enter the port number of the monitoring agent. The default port is 10088.

8 Click the **Apply** button.

▼ **To configure MediaBridge monitoring for QuickTime manually:**

1 In a text editor, open the `records.config` file located in the `config` directory.

2 Edit the following variables:

Variable	Description
proxy.config.qt.media_bridge.name	Set this variable to specify the name of the MediaBridge node.
proxy.config.qt.media_bridge.port	Set this variable to specify the port of the MediaBridge node. The default port is 10036.
proxy.config.qt.media_bridge.mount_point	Set this variable to specify the mount point for MDN streams.
proxy.config.qt.media_bridge.monitor.name	Set this variable to specify the name of the monitoring agent.
proxy.config.qt.media_bridge.monitor.port	Set this variable to specify the port number of the monitoring agent. The default port is 10088.

3 Save and close the `records.config` file.

4 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

5 Run the command `traffic_line -x` to apply the configuration changes.

Configuring RealProxy Options

This section discusses the following topics:

Changing the Default Port for Explicit Requests

By default, Traffic Edge uses port 9231 to contact RealProxy to serve explicit proxy requests from Real media players. To change the port number, you must edit a configuration file manually. Use the following procedure.

▼ **To change the port Traffic Edge uses to contact RealProxy:**

1. In a text editor, open the `records.config` file located in the `config` directory.
2. Edit the following variable:

Variable	Description
proxy.config.rni.proxy_port	Set this variable to specify the port that Traffic Edge uses to contact RealProxy to serve explicit proxy requests from Real media players.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

Setting Up RealProxy Tunneling

In transparent proxy mode, Traffic Edge receives both QuickTime and Real media player requests on the same RTSP port. By default, Traffic Edge sends the Real media player a redirect to port 9231 on which RealProxy is listening.

If you are concerned about exposing the Traffic Edge IP address to the network, you can configure Traffic Edge to establish a tunnel to the Traffic Edge RealProxy port (typically port 9231). In this case, Traffic Edge tunnels control messages and data to clients from RealProxy, avoiding the redirect message.

Note Traffic Edge uses RealProxy tunneling automatically for RealOne Player requests because RealOne Players do not fully support RTSP redirection.

▼ **To set up RealProxy tunneling:**

1. In a text editor, open the `records.config` file located in the `config` directory.
2. Add the following line at the end of the file:

   ```
   CONFIG proxy.config.qt.tunnel_rni_req INT 1
   ```
3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -L` to restart Traffic Edge on the local node or `traffic_line -M` to restart Traffic Edge on all the nodes in a cluster.

Configuring the RealProxy Restart Limit

You can configure the amount of time that `traffic_cop` service waits before restarting RealProxy either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ **To configure the RealProxy restart limit from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Streaming Media** button and then click the **Real Networks** button.
3. In the **RealProxy Restart Limit** field, enter the number of seconds that the `traffic_cop` service waits before restarting RealProxy. The default value is 20 seconds.
4. Click the **Apply** button.

▼ **To configure the RealProxy restart limit manually:**

1. In a text editor, open the `records.config` file located in the `config` directory.
2. Edit the following variable:

Variable	Description
proxy.config.rni.proxy_restart_interval	Set this variable to specify the number of seconds the `traffic_cop` service waits before restarting RealProxy. The default value is 20 seconds.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

Configuring Passthrough

In passthrough mode, Traffic Edge passes requested streams from the origin server to the client. Traffic Edge opens new control and data connections to the origin server for each client request.

You can enable Traffic Edge to redirect Real media player requests (determined by the RTSP headers) to a proxy that can pass through the RTSP requests to the origin server. Traffic Edge uses the Real Networks passthrough proxy for RTSP to enable QuickTime to redirect the requests. (For detailed information about the Real Networks passthrough proxy, see the Real Networks firewall proxy administration kit.)

Note To configure a Layer-4 switch for passthrough mode, do not set up a redirection rule for PNA requests to port 7070. PNA requests bypass Traffic Edge automatically without special configuration.

▼ **To configure passthrough for Real Networks:**

1. Copy the publicly available Real Networks passthrough proxy executable (`rtspd`) to a destination directory.
2. In a text editor, open the Traffic Edge `records.config` file.
3. Set the variable `proxy.config.rni.proxy_port` to 9231.

   In transparent mode, the Traffic Edge ARM sends port 554 traffic to the QuickTime plugin and the QuickTime plugin sends Real media player requests to the Real Networks passthrough proxy at this port. There is no ARM redirection rule for port 7070.
4. Add the following line at the end of the `records.config` file to specify the exact command line (including the pathname) that the `start_traffic_server` command uses to execute the Real Networks passthrough proxy:

   ```
   CONFIG proxy.config.rni.rpass_restart_cmd STRING /home/inktomi/
   1.5.0/rtspd/rtspd -p 9231
   ```
5. Save and close the `records.config` file.
6. Navigate to the `bin` directory and enter the following command to stop Traffic Edge:

   ```
   ./stop_traffic_server
   ```
7. Type the following command to set a parent value for the passthrough proxy:

   ```
   /rtspd -p 9231 -f parentproxy.inktomi.com:9231
   ```
8. Enter the following command to start Traffic Edge:

   ```
   ./start_traffic_server
   ```

Using Multicast

Traffic Edge can multicast streams to Real media players. Typically, when Traffic Edge splits an incoming stream into multiple outgoing streams, the transport protocol is UDP and so Traffic Edge needs to write UDP packets to the IP address of each client. If the clients can be reached via multicast, Traffic Edge can send out only one stream of multicast packets to the multicast address.

Traffic Edge supports multicast automatically; you do not have to perform any special Traffic Edge configuration. However, for a Real media player to receive multicast packets, the multicast option on the Real media player must be enabled (the multicast option *is* enabled by default, but it can be disabled by the client). Refer to the Real Networks documentation for additional information.

Very few routers pass multicast packets.

Configuring WMT Options

This section discusses the following topics:

Changing the MMS Proxy Port

By default, Traffic Edge uses port 1755 to receive MMS requests from a Windows Media Player. You can change the port either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

Note In reverse proxy mode, Traffic Edge rewrites the WMT metafile using the port specified in the `remap.config` file. To use a different port for reverse proxy caching, specify the new port in the `reverse_remap` rule.

▼ To change the MMS proxy port from Traffic Manager:

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Streaming Media** button and then click the **Windows Media** button.
3. In the **Windows Media Proxy Port** field, enter the port number Traffic Edge uses to receive MMS requests from a Windows Media Player.
4. Click the **Apply** button.

▼ To change the MMS proxy port manually:

1. In a text editor, open the `records.config` file located in the `config` directory.
2. Edit the following variable:

Variable	Description
proxy.config.wmt.port	Set this variable to specify the port number Traffic Edge uses to receive MMS requests from a Windows Media Player.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

Disabling ASX File Rewrite

The ASX file rewrite option enables Traffic Edge to rewrite the media content URL in a metafile to point to Traffic Edge instead of the Windows Media Server.

By default, the ASX rewrite option is enabled. Disable the ASX rewrite option only if you are certain that all your clients use Windows Media Player 7.0 and above. You can disable the ASX file rewrite option either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ **To disable the ASX file rewrite option from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Streaming Media** button and then click the **Windows Media** button.
3. Disable the **ASX Rewrite** option.
4. Click the **Apply** button.

▼ **To disable the ASX file rewrite option manually:**

1. In a text editor, open the `records.config` file located in the `config` directory.
2. Edit the following variable:

Variable	Description
proxy.config.wmt.asx_rewrite.enabled	Set this variable to 0 to disable the ASX file rewrite option.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

Configuring Memory for WMT Retransmissions

Inktomi recommends that you allocate more memory to store data to reply to retransmission requests if you expect Windows Media Players to use UDP to communicate with Traffic Edge or if you experience network congestion.

You can configure the amount of memory Traffic Edge allocates for retransmission requests from clients either by using Traffic Manager or by editing a configuration file manually. Both procedures as provided below.

▼ **To configure memory for WMT retransmissions from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Streaming Media** button and then click the **Windows Media** button.

3 In the **Maximum Memory Size** field of the **Retransmit Window** area, enter the amount of memory Traffic Edge uses to store data to reply to retransmission requests. The default value is 20971520 bytes (20 MB).

4 Click the **Apply** button.

▼ **To resize memory usage for WMT retransmissions manually:**

1 In a text editor, open the `records.config` file located in the `config` directory.

2 Edit the following variable:

Variable	Description
proxy.config.wmt.max_rexmit_memory	Set this variable to specify the amount of memory Traffic Edge uses to store data to reply to retransmission requests. The default value is 20971520 bytes (20 MB).

3 Save and close the `records.config` file.

4 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

5 Run the command `traffic_line -x` to apply the configuration changes.

Using WMT Media Push

The WMT media push option lets you preload WMT media files into the Traffic Edge cache to reduce the traffic load to an origin server; for example, you can use this option if the link to the origin server is slow or expensive, such as satellite link.

Note You can use the WMT media push option only for WMT files such as `.asf`, `.wma`, and `.wmv` files.

IMPORTANT To use WMT media push, you must install WMT media push support during Traffic Edge installation. The installation script installs a plugin (`mixt-push.so`) in the Traffic Edge `config/plugins/mixt` directory and a push program (`pushfile`) in the Traffic Edge `bin` directory.

To preload WMT media files into the Traffic Edge cache, perform the following tasks:

- Enable the media push option on the Traffic Edge node; refer to *Enabling the WMT Media Push Option, on page 77*.
- Configure metadata parameters (optional); refer to *Configuring Metadata Parameters, on page 78*.
- Preload WMT media files into the Traffic Edge cache; refer to *Preloading WMT Files into the Cache, on page 78*.

As additional tasks, you can display statistics about the WMT file you preloaded; refer to *Displaying the Status of Preloaded WMT Files, on page 79* and delete preloaded files from the Traffic Edge cache; refer to *Deleting Preloaded WMT Files, on page 80*.

Enabling the WMT Media Push Option

You can enable the WMT media push option either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ **To enable the WMT media push option from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Streaming Media** button and then click the **Windows Media** button.
3. Click the **Media Push** tab.
4. Enable the **Media Push** option.

   This option is enabled by default when you install media push during Traffic Edge installation.
5. In the **Port** field, enter the port you want to use for media push on the Traffic Edge node. The default value is 1900.
6. In the **Password** field, enter the password used for authentication. The value can be any ASCII string and cannot contain spaces. The default value is `NULL`, which disables password authentication.
7. Click the **Apply** button.

▼ **To enable the WMT media push option manually:**

1. In a text editor, open the `records.config` file located in the `config` directory.
2. Edit the following variables:

Variable	Description
proxy.config.mixt.push.enabled	Set this variable to 1 to enable the WMT media push option. This variable is enabled by default when you install media push during Traffic Edge installation.
proxy.config.mixt.push.port	Set this variable to specify the port you want to use for media push on this Traffic Edge node. The default value is 1900.
proxy.config.mixt.push.password	Set this variable to specify the password used for authentication. The value can be any ASCII string and cannot contain spaces. The default value is `NULL`, which disables password authentication.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

Configuring Metadata Parameters

For each WMT file that you preload into the cache, you can create metadata that specifies various caching parameters or customer tags. You can retrieve and inspect this metadata later, as described in *Displaying the Status of Preloaded WMT Files, on page 79*.

Traffic Edge provides a sample file `metadata-sample.xml` that you can edit to specify your metadata. The file is located in the Traffic Edge `config/plugins/mixt` directory. To create metadata for a WMT file, create an XML file that includes the push tag shown below.

```
<push>
   <pin days="days" hours="hours" minutes="minutes"/>
   <extra string="string"/>
</push>
```

The `pin` subtag specifies how long the content is stored in the cache to protect it from garbage collection.

The `extra` subtag specifies arbitrary information to associate with the preloaded content. Traffic Edge provides this information when you use the `stat` command, described in *Displaying the Status of Preloaded WMT Files, on page 79*. You can specify multiple `extra` subtags.

The following example stores a WMT stream in the cache for one day, two hours, and three minutes:

```
<push>
    <pin days="1" hours="2" minutes="3"/>
    <extra abc="defg"/>
</push>
```

IMPORTANT Specifying metadata is optional; however, a metadata file must exist for Traffic Edge to be able to preload files into the cache. This file can be empty. If the stream is not pinned in the cache, it is subject to normal garbage collection policies.

Preloading WMT Files into the Cache

After you have enabled the WMT media push option on a Traffic Edge node and have configured metadata parameters (optional), you can preload WMT files into the Traffic Edge cache. Use the following procedure.

▼ **To preload WMT files:**

1. Either transfer the WMT file that you want to preload to the filesystem on the Traffic Edge node or copy the `pushfile` program located in the Traffic Edge `config/plugins/mixt` directory to a remote host. The remote host must be able to access the WMT files you want to preload either via the local filesystem or a network-mounted filesystem.
2. From the Traffic Edge node or the remote host, enter the following command:

   `pushfile [-password password] -push host port URL metadata_file content_file [media_type]`

Optional

password is the password used for authentication. This must be the same password that is specified in the **Password** field under **Configure**/**Streaming Media**/**Windows Media**/**Media Push** in Traffic Manager (`proxy.config.mixt.push.password` in the `records.config` file).

host is the hostname or IP address of the Traffic Edge node on to which you want to preload the WMT file.

port is the port used for media push on the Traffic Edge node. The default port is 1900.

URL is the URL of the origin server. Traffic Edge does not contact the origin server for the content. The content is read from the `content_file` parameter.

metadata_file is the pathname of a metadata file in XML format. See *Configuring Metadata Parameters, on page 78*. If you have not created a metadata file, you can specify the sample metadata file that Traffic Edge provides in the `config/plugins/mixt` directory. This sample files provides comments only and does not specify any pinning parameters.

content_file is the pathname of the WMT file to preload into the cache.

Optional

media_type is the media type of the *content_file* parameter. In most cases, the media type is determined automatically from the URL. The only valid value is `WMT`.

The following example preloads the file `/home/data/howto.asf` into the cache on the Traffic Edge node `ts1`:

```
pushfile -password secret -push ts1 1900 mms://server1/howto.asf/home/
metadata.xml /home/data/howto.asf WMT
```

Displaying the Status of Preloaded WMT Files

You can display statistics about a preloaded WMT file, which includes the information you specified in the metadata file. To display the status of a WMT file that you have preloaded into the cache, enter the following command:

```
pushfile [-password password] -stat host port URL
```

Optional

password is the password used for authentication. This must be the same password that is specified in the **Password** field under **Configure**/**Streaming Media**/**Windows Media**/**Media Push** in Traffic Manager (`proxy.config.mixt.push.password` in the `records.config` file).

host is the hostname or IP address of the Traffic Edge node that contains the preloaded file. You must use the same format used when you preloaded the file.

port is the port used for media push on the Traffic Edge node.

URL is the exact URL specified to preload the file.

You can only display statistics for files that you have preloaded into the cache.

If you use the `stat` command immediately after preloading a file into the cache, Traffic Edge displays misleading statistics (Traffic Edge does not perform cache writes immediately but buffers files and writes them to disk periodically to increase performance).

Deleting Preloaded WMT Files

To delete a WMT file that has been preloaded into the Traffic Edge cache, type the following command:

```
pushfile [-password password] -delete host port URL
```

Optional

password is the password used for authentication. This must be the same password that is specified in the **Password** field under **Configure**/**Streaming Media**/**Windows Media**/**Media Push** in Traffic Manager (`proxy.config.mixt.push.password` in the `records.config` file).

host is the hostname or IP address of the Traffic Edge node that contains the preloaded file. You must use the same format used when you preloaded the file.

port is the port used for media push on the Traffic Edge node.

url is the exact URL specified to preload the file.

You can delete only WMT files that you preloaded into the cache. If you attempt to delete a file that has not been preloaded, you receive a `404 (Not Found)` error.

IMPORTANT

The `-delete` option does not actually delete the file from the cache but unpins it so that the entire file or its constituent blocks are subject to normal cache garbage-collection policies.

Using WMT Multicast

You can use the WMT multicast option to multicast WMT streams to clients.

Using multicast rather than unicast to stream media data between Traffic Edge and clients achieves greater efficiency. Typically, when Traffic Edge splits an incoming stream into multiple outgoing streams, the transport protocol is UDP and so Traffic Edge needs to write UDP packets to the IP address of each client. If the clients can be reached via multicast, Traffic Edge can send out only one stream of multicast packets to the multicast address.

Very few routers pass multicast packets.

To use the WMT multicast option, you must perform the following tasks:

- Enable the WMT multicast option; refer to *Enabling the WMT Multicast Option, on page 81*, below.
- Create an NSC file that contains information that Traffic Edge needs to start the multicast broadcast and information the client requires to receive a multicast broadcast; refer to *Creating an NSC File, on page 81*.

Optional

- Create an ASX file that contains the URL to the NSC file. This enables the client to request the multicast broadcast from the browser; refer to *Creating an ASX File, on page 82*.
- Control the multicast broadcast; refer to *Controlling Multicast from Traffic Edge, on page 82*.

IMPORTANT

For clients to receive multicast streams, the Windows Media Player must be multicast enabled. For information about enabling multicast on the Windows Media Player, refer to the Windows Media Player documentation.

Enabling the WMT Multicast Option

You can enable the WMT multicast option either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ **To enable the WMT multicast option from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Streaming Media** button and then click the **Windows Media** button.
3. Click the **Multicast** tab.
4. Enable the **Multicast** option.

   This option is enabled by default when you install WMT multicast during Traffic Edge installation.
5. Click the **Apply** button.

▼ **To enable the WMT multicast option manually:**

1. In a text editor, open the `records.config` file located in the `config` directory.
2. Edit the following variables:

Variable	Description
proxy.config.mixt.wmtmcast.enabled	Set this variable to 1 to enable the WMT multicast option. This option is enabled by default when you install WMT multicast during Traffic Edge installation.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

Creating an NSC File

Use the following procedure to create an NSC file.

▼ **To create an NSC file:**

1. Start Windows Media Administrator.
2. Create an NSC (`.nsc`) file that specifies the following information:
   - The multicast address and the port on which to multicast the stream
   - The unicast URL, which specifies how Traffic Edge obtains the stream and where the client is directed if it does not receive multicast packets after requesting the NSC file
   - The TTL (time-to-live), which specifies the extent of the multicast (through how many routers your multicast will travel before being dropped)

IMPORTANT Be careful when choosing the time-to-live. If the value is too large, the packets might travel on to other routers and could impact networks where there are no listeners. If the value is too small, the packets might not reach users on other networks. Also, the further a listener is from the multicast source, the worse the viewing experience can be due to loss of packets. Inktomi recommends a TTL value of less than 5.

Refer to the Windows Media Administrator documentation for more complete information about creating NSC files.

Creating an ASX File

If you want the client media player to be activated on an NSC request, you must create an ASX file that contains the URL to the NSC file. You must store the ASX file on a web server. For information about creating an ASX file, refer to the Windows Media Server documentation.

Controlling Multicast from Traffic Edge

To control multicast broadcasts from Traffic Edge, you use the `wmtmcastctl` script located in the Traffic Edge `bin` directory. Enter the following command either from the Traffic Edge node or from a remote host that can access Traffic Edge:

```
c -username login -password password -proxy host
[-proxyport port] -command [-no-origin] URL
```

`login` is the login you use to access Traffic Manager.

`password` is the passsword you use to access Traffic Manager.

Optional *`port`* is the server port used by Traffic Edge. The default port is 8080.

`host` is the Traffic Edge hostname.

`command` is one of the following:

`idle` instructs Traffic Edge to send packets every second to the multicast address and the port specified in the NSC file so that clients can tune in before the broadcast starts.

`start` instructs Traffic Edge to start sending the media data on the multicast address. You do *not* have to use the `idle` option before you start the broadcast.

You can use the optional parameter `-no-origin` with the `start` command if there is no media server or if your link to the origin server is slow or expensive: for example, you use a satellite link. For the `-no-origin` parameter to work, the stream must be preloaded into the cache and pinned for an adequate amount of time. If Traffic Edge cannot find the stream, it stops the multicast broadcast.

`stop` instructs Traffic Edge to stop sending the media data on the multicast address.

`URL` is the URL to the NSC file for the stream.

The following example instructs Traffic Edge to start a multicast broadcast:

```
wmtmcastctl -username admin -password secret -proxy proxy1 -start
http://webserver/file.nsc
```

Configuring MediaBridge Monitoring

If your Traffic Edge node is part of an Inktomi Media Distribution Network, you can configure MediaBridge monitoring for WMT either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below. For information about the Inktomi Media Distribution Network, refer to the Inktomi Media Distribution Network documentation.

▼ **To configure MediaBridge monitoring for WMT from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Streaming Media** button and then click the **Windows Media** button.
3. Click the **Media Bridge** tab.
4. In the **Media Bridge Name** field, enter the name of the MediaBridge node.
5. In the **Media Bridge Port** field, enter the port number of the MediaBridge node. The default port is 10022.
6. In the **Media Bridge Mount Point** field, enter the mount point for MDN streams.
7. In the **Monitor Name** field, enter the name of the monitoring agent.
8. In the **Monitor Port** field, enter the port number of the monitoring agent. The default port is 10088.
9. Click the **Apply** button.

▼ **To configure MediaBridge monitoring for WMT manually:**

1. In a text editor, open the `records.config` file located in the `config` directory.
2. Edit the following variables:

Variable	Description
proxy.config.wmt.media_bridge.name	Set this variable to specify the name of the MediaBridge node.
proxy.config.wmt.media_bridge.port	Set this variable to specify the port of the MediaBridge node. The default port is 10022.
proxy.config.wmt.media_bridge.mount_point	Set this variable to specify the mount point for MDN streams.
proxy.config.wmt.media_bridge.monitor.name	Set this variable to specify the name of the monitoring agent.
proxy.config.wmt.media_bridge.monitor.port	Set this variable to specify the port number of the monitoring agent. The default port is 10088.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

CHAPTER 5

Explicit Proxy Caching

If you want to use Traffic Edge as an explicit proxy cache, you must configure client software (for example, browsers and media players) to send requests directly to Traffic Edge.

This chapter discusses the following topics:

Explicit Proxy Caching for HTTP

If you do not configure Traffic Edge to use the transparency option (with which client requests are intercepted on the way to origin servers by a switch or router and rerouted to the Traffic Edge machine), clients must configure their web browsers to send HTTP requests to the Traffic Edge proxy cache. Clients can configure their web browsers in one of three ways:

- By manually configuring their browsers to send requests directly to Traffic Edge; refer to *Configuring Browsers Manually, on page 85*.
- By configuring their browsers to download proxy configuration instructions from a PAC (Proxy Auto-Configuration) file; refer to *Using a PAC File, on page 86*.
- By using WPAD (Web Proxy Auto-Discovery) to automatically download proxy configuration instructions from a WPAD server (Microsoft Internet Explorer only); refer to *Using WPAD, on page 88*.

Configuring Browsers Manually

To manually configure a browser to send HTTP requests to Traffic Edge, clients must provide the following information:

- The fully qualified hostname or IP address of the Traffic Edge node
- The Traffic Edge proxy server port (port 8080)

In addition, clients can specify not to use Traffic Edge for certain sites; requests to the listed sites go directly to the origin server.

The procedures for manual configuration vary among browsers; for example, for Netscape Navigator Version 4.7, the proxy configuration settings are in the **Edit** menu under **Preferences/Advanced/Proxies**. For Microsoft Internet Explorer Version 5.0, the proxy configuration settings are in **Tools/Internet Options/Connections/LAN Settings**. By default, Microsoft Internet Explorer sets all protocols to the same proxy server. Refer to the browser documentation for complete proxy configuration instructions.

You do not have to set any special configuration options on Traffic Edge to accept requests from manually configured browsers.

Using a PAC File

A PAC file is a specialized JavaScript function definition that a browser calls to determine how requests are handled. Clients must specify the URL from which the PAC file is loaded in their browser settings.

You can store a PAC file on Traffic Edge and provide the URL for this file to your clients.

Note The PAC file does *not* have to reside on the Traffic Edge system. It can reside on any server in your network.

If you want to store a PAC file on the Traffic Edge system, you must perform the following configuration:

- Either copy an existing PAC file into the Traffic Edge `config` directory or enter a script that defines the proxy server configuration settings in the `proxy.pac` file provided (the file is empty by default).
- Specify the port that Traffic Edge uses to serve the PAC file. The default port is 8083.

You can configure Traffic Edge to provide a PAC file either by using Traffic Manager or by editing configuration files manually. Both procedures are provided below.

▼ **To configure Traffic Edge to provide a PAC file from Traffic Manager:**

1. If you have an existing PAC file, replace the `proxy.pac` file located in the Traffic Edge `config` directory with the existing file.
2. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
3. On the **Configure** tab, click the **Content Routing** button and then click the **Browser Auto-Config** button.
4. In the **Auto-Configuration Port** field of the **PAC** tab, specify the port that Traffic Edge uses to serve the PAC file. The default port is 8083.
5. The **PAC Settings** area displays the `proxy.pac` file:
   - If you copied an existing PAC file into the Traffic Edge `config` directory, the `proxy.pac` file contains your proxy configuration settings. Check the settings and make changes if necessary.
   - If you did not copy an existing PAC file into the Traffic Edge `config` directory, the `proxy.pac` file is empty. Enter the script that provides the proxy server configuration settings. A sample script is provided in *Sample PAC File, on page 87*.
6. Click the **Apply** button.
7. Restart Traffic Edge by clicking the **Restart** button on the **Configure**/**My Proxy**/**Basic/General** tab.
8. Inform your users to set their browsers to point to this PAC file; for example, if the PAC file is located on Traffic Edge with the hostname `proxy1` and Traffic Edge uses the default port 8083 to serve the file, users must specify the following URL in the proxy configuration settings:

   ```
   http://proxy1.company.com:8083/proxy.pac
   ```

Note The procedures for specifying the PAC file location vary among browsers; for example, for Netscape Navigator Version 4.7, you set the location of the PAC file in the **Automatic proxy configuration** field under **Edit/Preferences/Advanced/Proxies**. For Microsoft Internet Explorer Version 5.0, you set the location of the PAC file in the **Use automatic configuration script** field under **Tools/Internet Options/Connections/LAN Settings**. Refer to the browser documentation for complete proxy configuration instructions.

▼ To configure Traffic Edge to provide a PAC file manually:

1. If you have an existing PAC file, replace the `proxy.pac` file located in the Traffic Edge `config` directory with the existing file.
2. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
3. Edit the following variable:

Variable	Description
proxy.config.admin.autoconf_port	Set this variable to specify the port that Traffic Edge uses to serve the PAC file. The default port is 8083.

4. Save and close the `records.config` file.
5. In a text editor, open the `proxy.pac` file located in the Traffic Edge `config` directory.
   - If you copied an existing PAC file into the Traffic Edge `config` directory, the `proxy.pac` file contains your proxy configuration settings. Check the settings and make changes if necessary.
   - If you did not copy an existing PAC file into the Traffic Edge `config` directory, the `proxy.pac` file is empty. Enter a script that will provide the proxy server configuration settings. A sample script is provided in *Sample PAC File*, below.
6. Save and close the `proxy.pac` file.
7. Restart Traffic Edge.
8. Inform your users to set their browsers to point to this PAC file; refer to *step 8, on page 86*.

Sample PAC File

The following sample PAC file instructs browsers to connect directly to all hosts without a fully qualified domain name and to all hosts in the local domain. All other requests go to the Traffic Edge called `myproxy.company.com`.

```
function FindProxyForURL(url, host)
{
       if (isPlainHostName(host)) ||
       (localHostOrDomainIs(host, ".company.com")) {
        return "DIRECT";
       }
       else
       return "PROXY myproxy.company.com:8080;" +
              "DIRECT";
}
```

Using WPAD

WPAD allows Internet Explorer Version 5 and later to automatically detect a server that will supply it with proxy server configuration settings. Clients do not have to perform any browser configuration to send requests to a proxy server: a single server provides the settings to all clients on the network.

When an Internet Explorer Version 5 browser starts up, it searches for a WPAD server that will supply it with proxy server configuration settings. It adds the hostname `wpad` to the current fully qualified domain name; for example, a client in `x.y.company.com` searches for a WPAD server at `wpad.x.y.company.com`. If unsuccessful, the browser removes the bottommost domain and tries again; for example, it tries `wpad.y.company.com`. The browser stops searching when it detects a WPAD server or reaches the third-level domain, `wpad.company.com`. The algorithm stops at the third level so that the browser does not search outside the current network.

Note By default, Microsoft Internet Explorer Version 5 and later is set to automatically detect WPAD servers. However, browser users can disable this setting.

You can configure Traffic Edge to be a WPAD server so that it can serve WPAD autoconfiguration files. To configure Traffic Edge as a WPAD server, you must:

- Either copy an existing `wpad.dat` file into the Traffic Edge `config` directory or enter a script that provides the proxy server configuration settings in the `wpad.dat` file provided. The file is empty by default (refer to the following procedure).
- Add a special remap rule in the Traffic Edge `ipnat.conf` file and enable the ARM (transparency) option (refer to the following procedure).

IMPORTANT To serve WPAD autoconfiguration files, Traffic Edge must use the ARM driver. The ARM driver is installed automatically during installation.

You can configure Traffic Edge to be a WPAD server either by using Traffic Manager or by editing configuration files manually. Both procedures are provided below.

▼ To configure Traffic Edge as a WPAD server from Traffic Manager:

1. If you have an existing `wpad.dat` file, replace the `wpad.dat` file located in the Traffic Edge `config` directory with your existing file.
2. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*
3. On the **Configure** tab, click the **Content Routing** button and then click the **Browser Auto-Config** button.
4. Click the **WPAD** tab to display the `wpad.dat` file.
5. If you copied an existing `wpad.dat` file into the Traffic Edge `config` directory, the `wpad.dat` file contains your proxy configuration settings. Check the settings and make changes if necessary.

   If you did not copy an existing `wpad.dat` file into the Traffic Edge `config` directory, the `wpad.dat` file is empty. Enter a script that will provide the proxy server configuration settings. A sample script is provided in *Sample PAC File, on page 87* (a `wpad.dat` file can contain the same script as a `proxy.pac` file).
6. Click the **Apply** button to save your changes.
7. On the **Configure** tab, click the **My Proxy** button and then click the **Basic** button.
8. In the **Features** table, click the **ARM On** button in the **Networking** section.

9 Click the **Apply** button.

10 On the **Configure** tab, click the **Networking** button and then click the **ARM** button.

11 In the **Network Address Translation (NAT)** section, click the **Edit File** button to add a special remap rule to the `ipnat.conf` file.

The configuration file editor for the `ipnat.conf` file opens.

12 Enter information in the fields provided and then click the **Add** button:

- In the **Ethernet Interface** field, enter the network interface that receives browser WPAD requests: for example, `hme0` or `eth0`.
- From the **Connection Type** drop-down list, select **tcp**.
- In the **Source IP** field, enter the IP address of the Traffic Edge that will be resolved to the WPAD server name by the local name servers: for example, 123.456.7.8.
- In the **Source CIDR** field, enter 32
- In the **Source Port** field, enter 80.
- In the **Destination IP** field enter the same IP address you entered in the **Source IP** field.
- In the **Destination Port** field, enter 8083.

13 Click the **Apply** button to save your changes and then click the **Close** button to exit the configuration file editor.

14 Restart Traffic Edge by clicking the **Restart** button on the **Configure**/**My Proxy**/**Basic/General** tab.

▼ To configure Traffic Edge as a WPAD server manually:

1 From a text editor, open the `wpad.dat` file located in the Traffic Edge `config` directory.

2 If you copied an existing `wpad.dat` file into the Traffic Edge `config` directory, the `wpad.dat` file contains your proxy configuration settings. Check the settings and make changes if necessary.

If you did not copy an existing `wpad.dat` file into the Traffic Edge `config` directory, the `wpad.dat` file is empty. Enter a script that will provide the proxy server configuration settings. A sample script is provided in *Sample PAC File, on page 87* (a `wpad.dat` file can contain the same script as a `proxy.pac` file).

3 Save and close the `wpad.dat` file.

4 In a text editor, open the `ipnat.conf` file located in the Traffic Edge `config` directory.

5 Add the following special remap rule to the end of the file:

`rdr` *`interface ipaddress`*`/32 port 80 ->` *`ipaddress`* `port 8083 tcp`

`interface` is the network interface that receives browser WPAD requests: for example, `hme0` or `eth0` and *`ipaddress`* is the IP address of the Traffic Edge that will be resolved to the WPAD server name by local name servers.

6 Save and close the `ipnat.conf` file.

7 In a text editor, open the `records.config` file.

8 Edit the following variable:

Variable	Description
proxy.config.arm.enabled	Set this variable to 1 to enable ARM.

9 Save and close the `records.config` file.

10 Restart Traffic Edge.

Explicit Proxy Caching for Streaming Media

If you do not configure Traffic Edge in forward proxy mode to use the transparency option (with which requests are intercepted on the way to origin servers by a switch or router and rerouted to the Traffic Edge machine), clients must configure their media players to send streaming media requests to the Traffic Edge proxy cache.

This section describes:

- *Explicit Proxy Caching for QuickTime Requests*, below
- *Explicit Proxy Caching for Real Media Player Requests, on page 92*
- *Explicit Proxy Caching for WMT Requests, on page 93*

Explicit Proxy Caching for QuickTime Requests

Figure 6 illustrates how Traffic Edge serves QuickTime requests in explicit proxy caching mode.

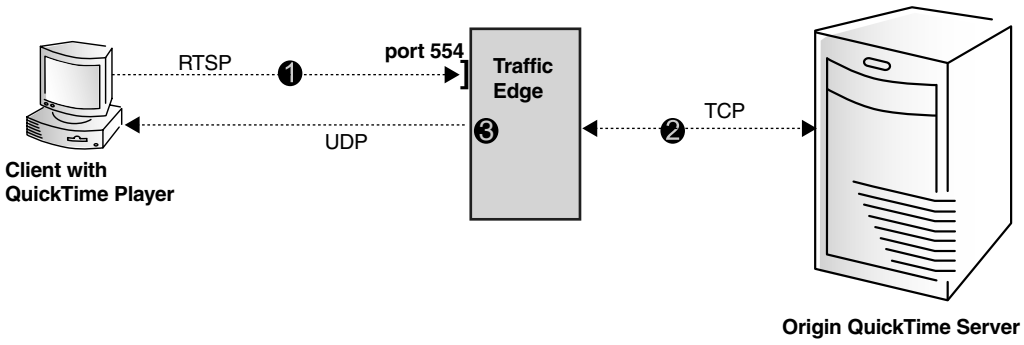


Figure 6 Explicit proxy caching for QuickTime requests

Figure 6 demonstrates the following steps:

1 A QuickTime Player client sends a request to the Traffic Edge RTSP port, typically port 554.

2 Traffic Edge opens a TCP data connection to the origin QuickTime server and obtains a last-modified time for the requested stream.

3 Traffic Edge determines if the requested stream is in the cache and if it is fresh (the last-modified time obtained from the QuickTime server must match the last-modified time in the cached stream). If so, Traffic Edge serves the requested stream; if not,

Traffic Edge retrieves the stream from the QuickTime server, stores the stream in the cache and simultaneously serves it to the client.

Configuring QuickTime Players

End users must configure their QuickTime Players to send requests directly to Traffic Edge for explicit proxy caching. The following procedure describes how to configure a QuickTime Player Version 5.0.

▼ **To configure QuickTime Player Version 5.0:**

1. From the QuickTime Player **Edit** menu, select **Preferences** and then select **QuickTime Preferences**.

   The QuickTime Settings dialog box opens.
2. From the drop-down list box, select **Streaming Proxy**.
3. Uncheck the **Use System Settings** check box.
4. Check the **RTSP Proxy Server** check box. In the field provided, enter the fully qualified domain name or the IP address of Traffic Edge. In the **Port ID** field, enter port number 554.
5. From the drop-down list box at the top of the QuickTime Settings dialog box, select **Streaming Transport**.
6. Check the **Use UDP, RTSP Port ID** check box and then select port **554**.
7. Close the QuickTime Settings dialog box.

Configuring QuickTime Servers

If you plan to connect Traffic Edge to a QuickTime Streaming Server, use the following procedure to enable Traffic Edge to cache the maximum percentage of streams. The procedure ensures that the origin server does not thin a stream when Traffic Edge is present in the path from the origin server to the player.

The following procedure describes how to configure an Apple Darwin Streaming Server Version 3.x.

▼ **To configure an Apple Darwin Streaming Server Version 3.x:**

1. In a text editor, open the `streamingserver.xml` file that is distributed with the QuickTime server.

   The `streamingserver.xml` file is typically located in the `/etc/streaming` directory and contains information about where to write log files, where to obtain the streaming media content, and where to find loadable modules.
2. Edit the following parameters in the `streamingserver.xml` file as shown below:

   ```
   <PREF NAME="tcp_max_video_delay_tolerance" TYPE="SInt32">500000</PREF>
   <PREF NAME="tcp_max_audio_delay_tolerance" TYPE="SInt32">500000</PREF>
   ```
3. Close and save the `streamingserver.xml` file.
4. Stop and restart the QuickTime server from the QuickTime server GUI.

5 Restart the QuickTime server and the modified `streamingserver.xml` file as follows:

```
./darwinstreamingserver -c /etc/streamingserver.xml
```

You must use an absolute path name for the `streamingserver.xml` file.

Explicit Proxy Caching for Real Media Player Requests

Figure 7 illustrates how Traffic Edge serves Real media player requests in explicit proxy caching mode. Traffic Edge supports explicit proxy caching for both RTSP and PNA requests.

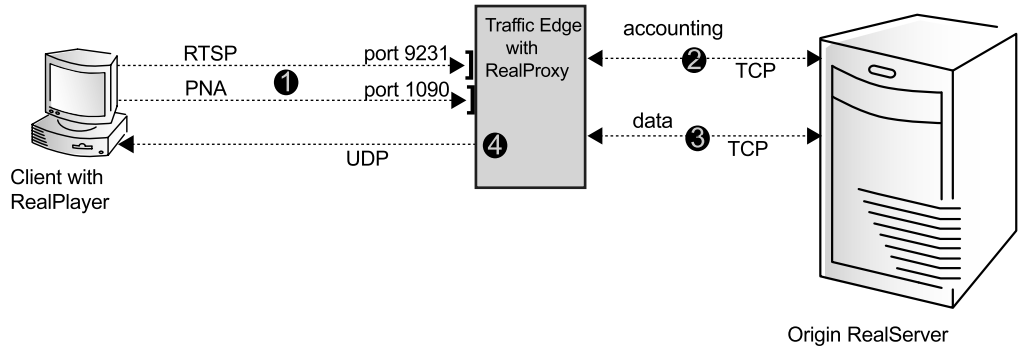


Figure 7 Explicit proxy caching for Real media player requests

Figure 7 demonstrates the following steps:

1 A Real media player client sends a request to the Traffic Edge RealProxy port, typically port 9231 for RTSP or port 1090 for PNA.

2 RealProxy opens a TCP accounting connection on port 554 to the origin RealServer, dedicating the connection to the client that sent the request.

3 If the client is allowed access, RealProxy opens a TCP data connection to the origin RealServer. If not, RealProxy closes the accounting connection, returns an error to the client and the remaining steps do not apply.

4 RealProxy determines if the requested stream is in the cache and if it is fresh. If so, Traffic Edge serves the requested stream; if not, Traffic Edge retrieves the stream from the origin RealServer on port 7878, stores the stream in the cache and simultaneously serves it to the client.

Configuring Real Media Players

End users must configure their Real media players to send requests directly to Traffic Edge. The following procedure describes how to configure a RealPlayer Version 6.0.

▼ **To configure RealPlayer Version 6.0:**

1 From the RealPlayer **View** menu, select **Preferences**.

The Preferences dialog box opens.

2 Click the **Proxy** tab.

The proxy dialog box opens.

3 To send PNA requests directly to Traffic Edge, check the **Use PNA proxy** check box. In the field provided, enter the fully qualified domain name or the IP address of Traffic Edge. In the **Port** field, enter 1090. This is the port number that the RealPlayer uses to communicate with Traffic Edge.

4 Check the **Use RTSP Proxy** check box. In the field provided, enter the fully qualified domain name or the IP address of Traffic Edge. In the **Port** field, enter the port number that the RealPlayer uses to communicate with Traffic Edge. You must enter the same port number specified by the configuration variable `proxy.config.rni.proxy_port` in the Traffic Edge `records.config` file. The default port number is 9231.

5 Click the **OK** button.

Explicit Proxy Caching for WMT Requests

Figure 8 and *Figure 9* illustrate how Traffic Edge serves Windows Media requests in explicit proxy mode.

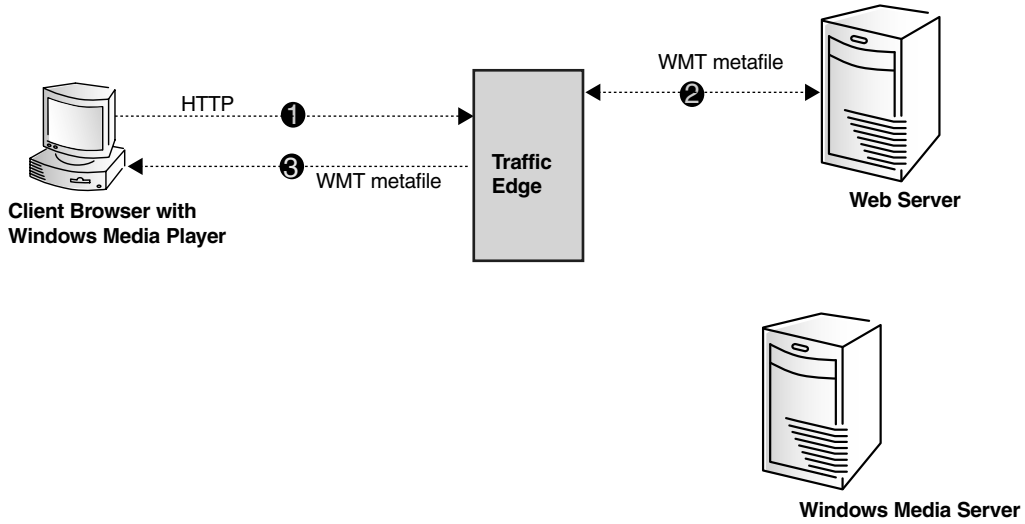


Figure 8 Explicit proxy caching for Windows Media requests

Figure 8 demonstrates the following steps:

1 A client browser sends a request for a WMT metafile directly to Traffic Edge (the browser is configured to send all HTTP requests to Traffic Edge on port 8080).

2 Traffic Edge obtains the WMT metafile from the web (HTTP) server.

3 Traffic Edge rewrites the contents of the metafile using the ASX rewrite option and then sends the metafile to the client. The rewritten metafile directs the media player to obtain the requested stream from Traffic Edge.

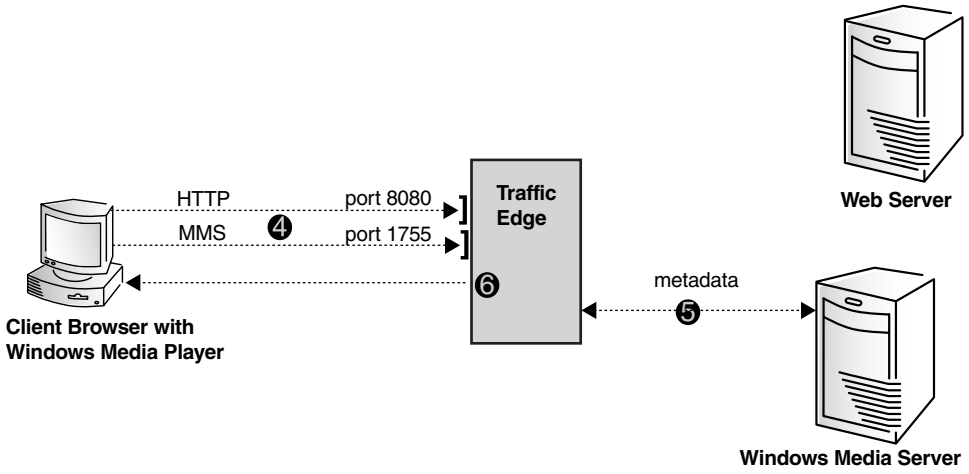


Figure 9 Explicit proxy caching for WMT, continued

4 The media player sends the rewritten media content URL to Traffic Edge via MMS on port 1755 or via HTTP on port 8080 (depending on the protocol specified in the URL).

5 Traffic Edge obtains metadata about the requested stream from the Windows Media Server.

6 Traffic Edge attempts to find an exact match between the Windows Media Server metadata and metadata in the Traffic Edge cache. If it does, the stream is a cache hit and Traffic Edge serves the requested stream to the client; if not, Traffic Edge retrieves the stream from the Windows Media Server, stores the stream in the cache and simultaneously serves it to the client.

IMPORTANT Traffic Edge can rewrite only URLs that appear in clear text in the HTML body of the ASX file. Traffic Edge cannot successfully rewrite ASX files implemented as javascript programs that construct the MMS URL from strings.

Note If you are *certain* that all your clients are running Windows Media Player Version 7.0 and above, you can disable the ASX rewrite option; refer to *Disabling ASX File Rewrite, on page 75*. With the ASX rewrite option disabled, Traffic Edge sends the requested ASX file to the browser without rewriting the URL. The media player can send the request for the URL in the metafile directly to Traffic Edge (Windows Media Player 7.0 and above can be configured to send WMT requests directly to Traffic Edge via HTTP or MMS).

Configuring Explicit Proxy Caching for WMT

For Traffic Edge to serve WMT requests in explicit proxy caching mode, clients must configure their web browsers to send HTTP requests for ASX files to the Traffic Edge proxy cache and configure their Windows Media Players (Version 7.0 and above) to send HTTP and/or MMS requests directly to the Traffic Edge proxy cache.

- To manually configure a browser to send HTTP requests to Traffic Edge, refer to *Configuring Browsers Manually, on page 85*.

- To manually configure a Windows Media Player to send HTTP requests to Traffic Edge, you must change the player proxy settings for HTTP to use Traffic Edge as the proxy server. The procedure in *To configure a Windows Media Player Version 7.0 for HTTP:* below describes how to configure a Windows Media Player Version 7.0.
- To manually configure a Windows Media Player to send MMS requests directly to Traffic Edge, you must change the player proxy settings for MMS to use Traffic Edge as the proxy server. The procedure in *To configure Windows Media Player Version 7.0 for MMS:* below describes how to configure a Windows Media Player Version 7.0.

Note Windows Media Players Version 6.x and earlier cannot be configured to send requests directly to a proxy.

▼ To configure a Windows Media Player Version 7.0 for HTTP:

1. Configure your browser to send HTTP requests to Traffic Edge; refer to *Configuring Browsers Manually, on page 85*.
2. From the Windows Media Player **Tools** menu, select **Options**.

   The Options dialog box opens.
3. Click the **Network** tab.
4. In the **Proxy Settings** area, double-click **HTTP**.

   The Configure Protocol dialog box opens.
5. Enable the **Use browser proxy settings** option and then click **OK**.
6. Click **OK** to close the Options dialog box.

▼ To configure Windows Media Player Version 7.0 for MMS:

1. From the Windows Media Player **Tools** menu, select **Options**.

   The Options dialog box opens.
2. Click the **Network** tab.
3. In the **Proxy Settings** area, double-click **MMS**.

   The Configure Protocol dialog box opens.
4. Enable the **Use the following proxy server** option.
5. In the **Address** field, enter the fully qualified hostname or IP address of the Traffic Edge node.
6. In the **Port** field, enter the port number. The default port number is 1755.
7. Click the **OK** button.

CHAPTER 6 Transparent Proxy Caching

The transparent proxy caching (transparency) option enables Traffic Edge to respond to Internet requests automatically by redirecting the traffic flow into the Traffic Edge cache after it has been intercepted by a Layer-4 (L4) switch or router.

In a nontransparent deployment, users must configure their browsers or media players to send web requests to the Traffic Edge proxy cache. Many sites have no direct control over a user's browser or media player settings, making it necessary for site administrators to provide configuration details.

This chapter discusses the following topics:

Nonstreaming Media Transparency

This section provides the following information:

- An overview of HTTP transparency; refer to *Transparency for HTTP Requests, on page 98*.
- An overview of FTP transparency; refer to *Transparency for FTP Requests, on page 99*.
- An overview of the steps needed to configure transparency for nonstreaming media; refer to *Configuring Traffic Edge for Nonstreaming Media Transparency, on page 100*.

Transparency for HTTP Requests

Figure 10 illustrates how Traffic Edge serves HTTP requests transparently:

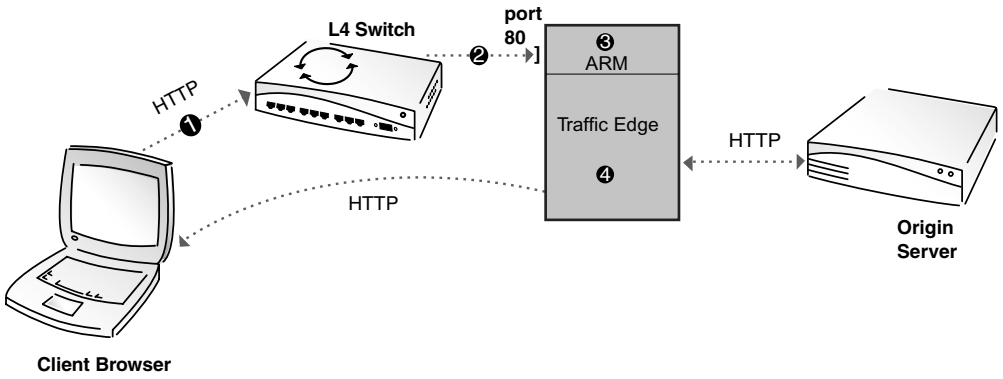


Figure 10 HTTP transparency

Figure 10 demonstrates the following steps:

1. A client browser sends an HTTP request addressed to the origin server on port 80.
2. A Layer-4 switch intercepts port 80 traffic and reroutes HTTP requests to the Traffic Edge machine.

   There are several ways to deploy Traffic Edge so that interception can take place; for example, you can use a WCCP-enabled router. Refer to *Interception Strategies, on page 109*, for details.
3. The Inktomi Adaptive Redirection Module (ARM) changes the destination IP address of an incoming packet to the Traffic Edge IP address and its destination port to the Traffic Edge proxy port (typically port 8080). Refer to *Enabling the ARM Option, on page 107*, for details.
4. Traffic Edge receives and begins processing the intercepted client requests as usual. If a request is a cache hit, Traffic Edge serves the requested object; if not, Traffic Edge retrieves the object from the origin server, stores the object in the cache and serves it to the client. On the way back to the client, the ARM changes the source IP address to the origin server IP address and the source port to the origin server port.

Traffic Edge can identify problem clients and servers dynamically, and the ARM can adaptively disable interception for these clients and servers, passing their traffic unimpeded to the origin server. In addition, you can create static rules to exempt clients

and servers from caching. Refer to *Interception Bypass (HTTP), on page 120*, for more information.

Transparency for FTP Requests

Figure 11 illustrates how Traffic Edge serves FTP requests transparently:

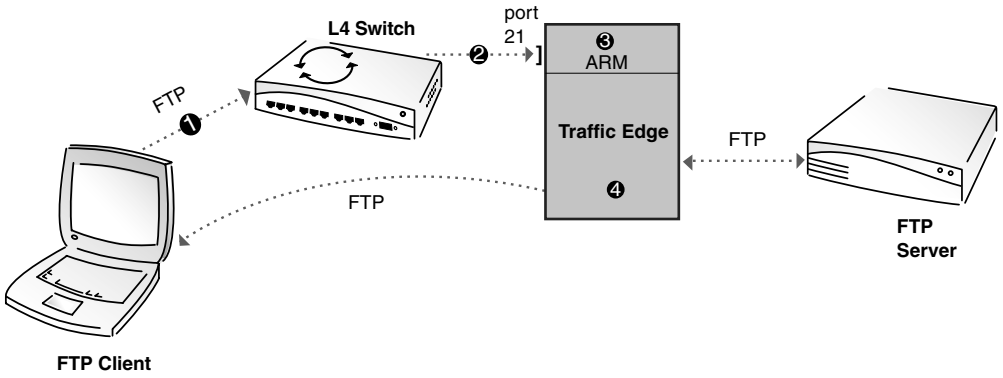


Figure 11 FTP transparency

Figure 11 demonstrates the following steps:

1. An FTP client sends an FTP request addressed to the FTP server on port 21.
2. A Layer-4 switch intercepts port 21 traffic and reroutes FTP requests to the Traffic Edge machine.

   There are several ways to deploy Traffic Edge so that interception can take place: for example, you can use a WCCP-enabled router. Refer to *Interception Strategies, on page 109*, for details.
3. The Inktomi Adaptive Redirection Module (ARM) changes the destination IP address of an incoming packet to the Traffic Edge IP address and its destination port to the Traffic Edge FTP proxy port (typically port 2121). Refer to *Enabling the ARM Option, on page 107*, for details.
4. Traffic Edge receives and begins processing the intercepted client requests as usual. If a request is a cache hit, Traffic Edge serves the requested object; if not, Traffic Edge retrieves the object from the FTP server, stores the object in the cache and serves it to the client. On the way back to the client, the ARM changes the source IP address to the FTP server IP address and the source port to the origin server port.

Configuring Traffic Edge for Nonstreaming Media Transparency

To configure Traffic Edge to serve nonstreaming media requests (HTTP and FTP) transparently, you must perform the following tasks:

- Enable the ARM module and specify redirection rules; refer to *Enabling the ARM Option, on page 107*.

  If you installed the ARM module during Traffic Edge installation, the installation program automatically enables the ARM and provides default redirection rules. You should check the redirection rules and make any necessary changes. Refer to *Enabling the ARM Option, on page 107*.

- Determine which transparency routing option you want to use. If you intend to use a Layer-4 switch or a WCCP router, you must configure the switch or router. You can obtain examples of Layer-4 switch and WCCP router configurations on the Inktomi Technical Support website: *http://support.inktomi.com*.

  If you intend to use a WCCP router, you must enable WCCP on Traffic Edge; refer to *Enabling WCCP on Traffic Edge, on page 111*.

Optional

- Configure Traffic Edge to reduce DNS lookups; refer to *Reducing DNS Lookups, on page 127*.

Optional

- Configure the ARM to bypass the proxy for certain requests (HTTP only); refer to *Interception Bypass (HTTP), on page 120*.

Optional

- Configure Traffic Edge to use the IP address of the client when it communicates with origin servers instead of its own IP address (HTTP only); refer to *IP Spoofing (HTTP), on page 127*.

Streaming Media Transparency

This section provides the following information:

- An overview of QuickTime transparency; refer to *Transparency for QuickTime Requests*, below.
- An overview of Real Networks transparency; refer to *Transparency for Real Media Player Requests, on page 102*.
- An overview of WMT transparency; refer to *Transparency for WMT Requests, on page 103*.
- An overview of the steps needed to configure transparency for streaming media; refer to *Configuring Traffic Edge for Streaming Media Transparency, on page 106*.

Transparency for QuickTime Requests

Figure 12 illustrates how Traffic Edge serves QuickTime requests transparently.

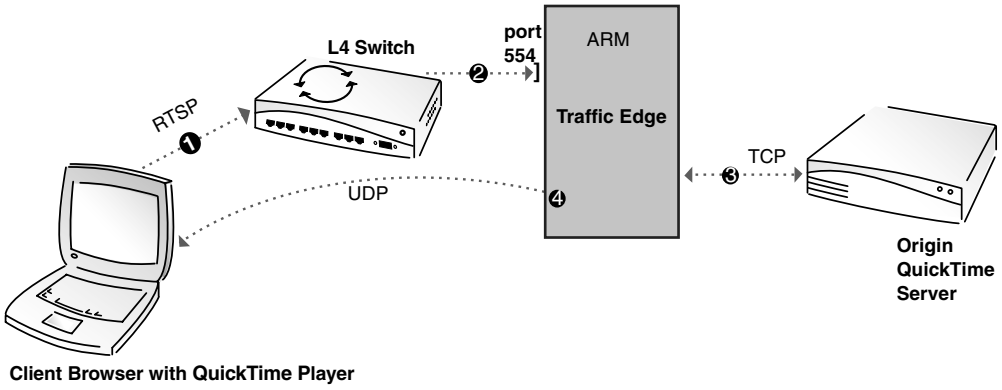


Figure 12 QuickTime transparency

Figure 12 demonstrates the following steps:

1. A QuickTime Player sends an RTSP request addressed to the origin QuickTime server on port 554.
2. The Layer-4 switch intercepts port 554 traffic and redirects requests to the Traffic Edge machine on port 554. The ARM changes the destination IP address of an incoming packet to the Traffic Edge IP address.

   There are several ways to deploy Traffic Edge so that interception can take place: for example, you can use a WCCP-enabled router. Refer to *Interception Strategies, on page 109*, for details.
3. Traffic Edge opens a TCP data connection to the origin QuickTime server and obtains a last-modified time for the requested stream.
4. Traffic Edge determines if the requested stream is in the cache and if it is fresh (the last-modified time obtained from the QuickTime server must match the last-modified time in the cached stream). If so, Traffic Edge serves the requested stream; if not, Traffic Edge retrieves the stream from the QuickTime server, stores the stream in the cache and simultaneously serves it to the client. On the way back to the client, the ARM changes the source IP address back to the QuickTime server IP address.

Transparency for Real Media Player Requests

Figure 13 illustrates how Traffic Edge serves Real media player RTSP requests transparently.

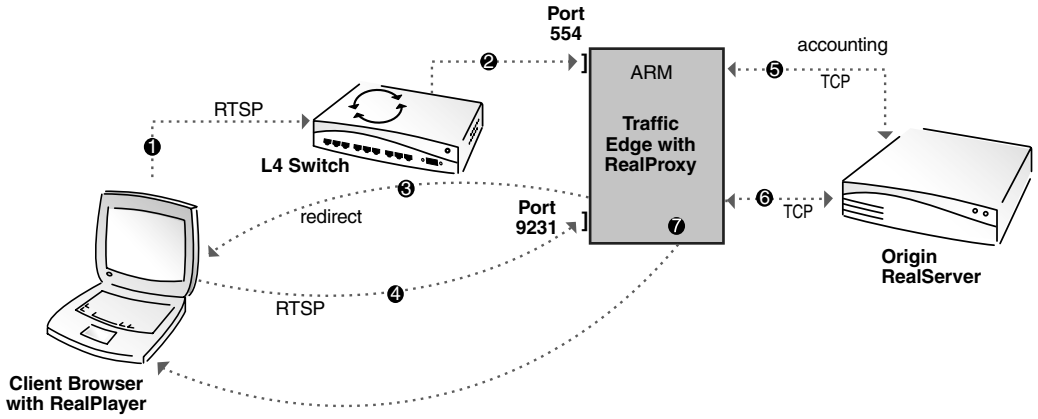


Figure 13 Real media player RTSP transparency

Figure 13 demonstrates the following steps:

1. A Real media player sends an RTSP request addressed to the origin RealServer on port 554.
2. The Layer-4 switch intercepts port 554 traffic and redirects requests to the Traffic Edge machine on port 554. The ARM changes the destination IP address of an incoming packet to the Traffic Edge IP address.

   There are several ways to deploy Traffic Edge so that interception can take place: for example, you can use a WCCP-enabled router. Refer to *Interception Strategies, on page 109*, for details.
3. Because the RTSP request comes from a Real media player, Traffic Edge issues the client a redirect to port 9231.

Note For RealOne Player requests, Traffic Edge does not issue a redirect to port 9231; instead, Traffic Edge establishes a tunnel to 9231 (the RealProxy port).

4. The Real media player sends the redirected RTSP request to the Traffic Edge RealProxy on port 9231.
5. RealProxy opens a TCP accounting connection on port 554 to the origin RealServer, dedicating the connection to the client that sent the request.
6. If the client is allowed access, RealProxy opens a TCP data connection to the origin RealServer. If not, RealProxy closes the accounting connection, returns an error to the client, and the remaining steps do not apply.
7. RealProxy determines if the requested stream is in the cache and if it is fresh. If so, RealProxy serves the stream to the client; if not, RealProxy retrieves the stream from the origin server on port 7878, stores the stream in the cache and simultaneously serves it to the client. On the way back to the client, the ARM changes the source IP address back to the origin RealServer IP address.

Traffic Edge transparency for PNA requests works in a similar way to transparency for RTSP requests. However, *step 1* through *step 4* differ in that Real media players send PNA

requests on port 7070 instead of port 554. The ARM redirects the requests to port 7272 and the Traffic Edge RealProxy issues the client a redirect to a port that is dynamically assigned for that particular request. The Real media player then sends the redirected PNA request to the dynamically assigned port.

Transparency for WMT Requests

Traffic Edge supports the following types of transparency for WMT:

- HTTP transparency provided by a Layer-4 switch or WCCP 2.0-enabled router to intercept browser requests for a WMT metafile. Traffic Edge rewrites the WMT metafile so that the Windows Media Player directs requests for WMT streams to Traffic Edge. See *Figure 14* and *Figure 15*.
- MMS and HTTP transparency provided by a Layer-4 switch or WCCP 2.0-enabled router to intercept Windows Media Player requests for a WMT stream. Traffic Edge does not have to rewrite WMT metafiles. See *Figure 16*.

IMPORTANT Traffic Edge can rewrite only URLs that appear in clear text in the HTML body of the ASX file. If you implement ASX files as javascript programs that construct the MMS URL from strings, you must configure MMS or HTTP transparency Windows Media Player requests.

Figure 14 and *Figure 15* illustrate HTTP transparency used to intercept a request for a WMT metafile. Using the ASX rewrite option, Traffic Edge rewrites the URL in the metafile to point to Traffic Edge.

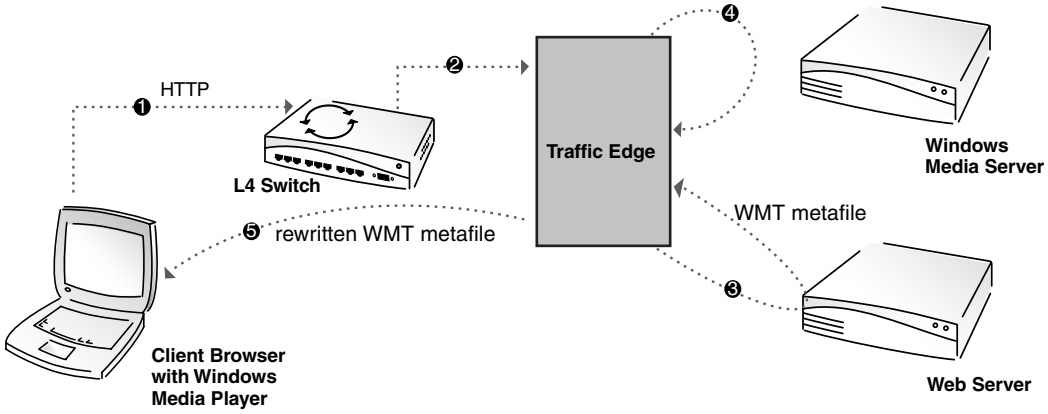


Figure 14 HTTP transparency to intercept a request for a WMT metafile

Figure 14 demonstrates the following steps:

1. The client browser sends a request for a WMT metafile to a web (HTTP) server on port 80.
2. A Layer-4 switch intercepts the HTTP request and redirects it to the Traffic Edge machine. The Inktomi Adaptive Redirection Module (ARM) changes the destination IP address of the incoming packet to the Traffic Edge IP address and the destination port to the Traffic Edge proxy port (typically port 8080).

   There are several ways to deploy Traffic Edge so that interception can take place: for example, you can use a WCCP-enabled router. Refer to *Interception Strategies, on page 109*, for details.
3. Traffic Edge obtains the WMT metafile from the web (HTTP) server.

4 Traffic Edge rewrites the URL in the WMT metafile using the ASX rewrite option.

5 Traffic Edge sends the metafile to the client (on the way back to the client, the ARM changes the source IP address of the packet back to the web server IP address and the port back to the HTTP port). The rewritten metafile directs Windows Media Player to obtain the stream from Traffic Edge.

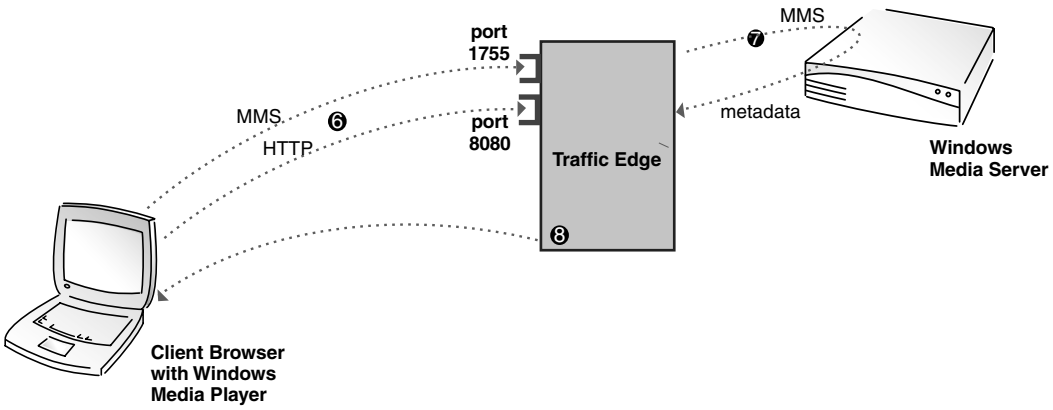


Figure 15 HTTP transparency to intercept a request for a WMT metafile, continued

Figure 15 demonstrates the following steps:

6 The Windows Media Player sends the rewritten media content URL to Traffic Edge via MMS on port 1755 or via HTTP on port 8080 (depending on the protocol specified in the URL).

7 Traffic Edge obtains metadata about the requested stream from the Windows Media Server.

8 Traffic Edge attempts to find an exact match between the origin WMT server metadata and metadata in the Traffic Edge cache. If it does, the stream is a cache hit and Traffic Edge serves the requested stream to the client; if not, Traffic Edge retrieves the stream from the Windows Media Server, stores the stream in the cache and simultaneously serves it to the client.

Figure 16 illustrates MMS transparency used to intercept requests from a Windows Media Player for a WMT stream. In this example, Traffic Edge does not need to rewrite metafiles (to disable the ASX file rewrite option, refer to *Disabling ASX File Rewrite, on page 75*).

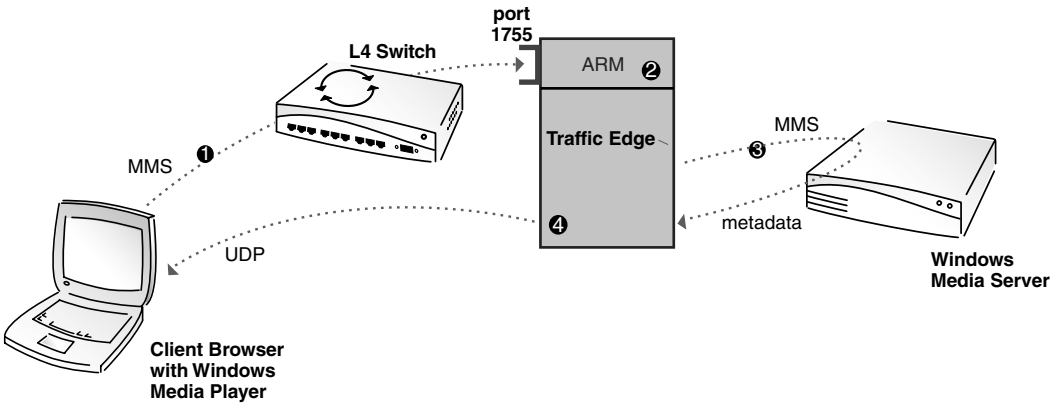


Figure 16 MMS transparency for WMT streams

Figure 16 demonstrates the following steps:

1. The Windows Media Player sends an MMS request addressed to the Windows Media Server on port 1755. The Layer-4 switch intercepts port 1755 traffic and redirects it to port 1755 on the Traffic Edge machine.

   There are several ways to deploy Traffic Edge so that interception can take place: for example, you can use a WCCP 2.0-enabled router. Refer to *Interception Strategies, on page 109*, for details.

2. The Inktomi Adaptive Redirection Module (ARM) changes the destination IP address of an incoming packet to the Traffic Edge IP address.
3. Traffic Edge obtains metadata about the stream from the origin WMT server.
4. Traffic Edge attempts to find an exact match between the origin WMT server metadata and metadata in the Traffic Edge cache. If it does, the stream is a cache hit and Traffic Edge serves the requested stream to the client; if not, Traffic Edge retrieves the stream from the Windows Media Server, stores the stream in the cache and simultaneously serves it to the client. On the way back to the client, the ARM changes the source IP address of the packet back to the Windows Media Server IP address.

HTTP transparency for Windows Media Player requests works in the same way as MMS transparency. However, the Media Player sends the request for the stream on the HTTP port (typically port 80). After the switch or router intercepts the request and forward it to the Traffic Edge machine, the ARM changes the IP address of the incoming packet to the Traffic Edge IP address and the port to the HTTP proxy port (typically 8080). On the way back to the client, the ARM changes the source IP address of the packet back to the Windows Media Server IP address and the port back to the HTTP port (80).

Configuring Traffic Edge for Streaming Media Transparency

To configure Traffic Edge to serve streaming media requests transparently, you must perform the following tasks:

- Enable the ARM module and specify redirection rules; refer to *Enabling the ARM Option, on page 107*.

Note If you installed the ARM module during Traffic Edge installation, the installation program automatically enables the ARM and supplies default redirection rules. You should check the redirection rules and make any necessary changes. Refer to *Enabling the ARM Option, on page 107*.

- Determine which transparency routing option you want to use. If you intend to use a Layer-4 switch or a WCCP router, you must configure the switch or router. You can obtain examples of Layer-4 switch and WCCP router configurations on the Inktomi Technical Support website: *http://support.inktomi.com*.

  If you intend to use a WCCP router, you must enable WCCP on Traffic Edge; refer to *Enabling WCCP on Traffic Edge, on page 111*.

- For Real Networks, determine if you need to enable RealProxy tunneling; refer to *Setting Up RealProxy Tunneling, on page 71*.

Enabling the ARM Option

The Traffic Edge *ARM* transparently inspects incoming packets before the IP layer sees them and readdresses the packets to Traffic Edge so that they can be served from the cache.

The ARM can make two changes to an incoming packet's address: its destination IP address and its destination port. For example, the destination IP address of an HTTP packet is readdressed to the IP address of Traffic Edge and the destination HTTP port is readdressed to the Traffic Edge HTTP proxy port (usually port 8080).

On the way back to the client, the ARM changes the source IP address to the origin server IP address and the source port to the origin server port.

The ARM component consists of several files and a kernel module, which are installed automatically during a Traffic Edge installation. The installation program also creates redirection rules to readdress packets using the IP address of the Traffic Edge machine and default port assignments.

For Traffic Edge to serve requests transparently, you must check the redirection rules in the `ipnat.conf` file and edit them if necessary. You must also enable the ARM option if you did not enable it during a custom Traffic Edge installation.

You can check redirection rules and enable the ARM option either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ **To check redirection rules and enable the ARM option from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. Click the **Configure** tab.

   The **General** tab of the **My Proxy**/**Basic** button displays.
3. In the **Features** table on the **General** tab, click the **ARM On** button in the **Networking** section.
4. Click the **Apply** button.
5. On the **Configure** tab, click the **Networking** button and then click the **ARM** button.
6. The **Network Address Translation (NAT)** section on the **General** tab displays the redirection rules in the `ipnat.conf` file. Check the redirection rules and make changes if necessary.
7. To change a redirection rule, click the **Edit File** button.

   The configuration file editor for the `ipnat.conf` file opens.
8. Select the rule you want to edit and then modify the appropriate fields. Click the **Set** button and then click the **Apply** button to apply your changes. Click the **Close** button to exit the configuration file editor.

   All fields are described in *Appendix B, Traffic Manager Configuration Options*.
9. Restart Traffic Edge by clicking the **Restart** button on the **Configure**/**My Proxy**/**Basic/General** tab.

▼ **To check redirection rules and enable the ARM option manually:**

1. In a text editor, open the `ipnat.conf` file located in the Traffic Edge `config` directory.
2. Check the redirection rules in the file and make changes if necessary. See examples that follow this procedure.
3. Save and close the `ipnat.conf` file.
4. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
5. Edit the following configuration variable:

Variable	Description
proxy.config.arm.enabled	Set this variable to 1 to enable the ARM option.

6. Save and close the `records.config` file.
7. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
8. Run the command `traffic_line -L` to restart Traffic Edge on the local node or `traffic_line -M` to restart Traffic Edge on all the nodes in a cluster.

The following sample output shows an HTTP redirect rule that instructs Traffic Edge to readdress all incoming HTTP packets to the Traffic Edge IP address (111.111.11.1) on the Traffic Edge HTTP proxy port 8080:

```
rdr hme0 0.0.0.0/0 port 80 -> 111.111.11.1 port 8080 tcp
```

The following sample output shows an FTP redirect rule that instructs Traffic Edge to readdress all incoming FTP packets to the Traffic Edge IP address (11.1.1.1) on the Traffic Edge FTP proxy port 2121:

```
rdr hme0 0.0.0.0/0 port 21 -> 11.1.1.1 port 2121 tcp
```

The following sample output shows an RTSP redirect rule that instructs Traffic Edge to readdress all incoming RTSP packets to the Traffic Edge IP address (11.1.1.1) on the Traffic Edge proxy port 554:

```
rdr hme0 0.0.0.0/0 port 554 -> 11.1.1.1 port 554 tcp
```

The following sample output shows an MMS redirect rule that instructs Traffic Edge to readdress all incoming MMS packets to the Traffic Edge IP address (11.1.1.1) on the Traffic Edge proxy port 1755:

```
rdr hme0 0.0.0.0/0 port 1755 -> 11.1.1.1 port 1755 tcp
```

Interception Strategies

Traffic Edge supports the following transparency routing solutions:

- A Layer-4 switch; refer to *Using a Layer-4 Switch*, below.
- A Cisco IOS-based router using the Web Cache Control Protocol (WCCP); refer to *Using a WCCP-Enabled Router, on page 110*.
- Policy-based routing; refer to *Using Policy-Based Routing, on page 118*.
- Software routing; refer to *Using Software-Based Transparency Solutions, on page 119*.

Using a Layer-4 Switch

Layer-4 switches can rapidly redirect supported protocols to Traffic Edge, while passing all other Internet traffic through directly to its destination. *Figure 17* illustrates this scenario for HTTP.

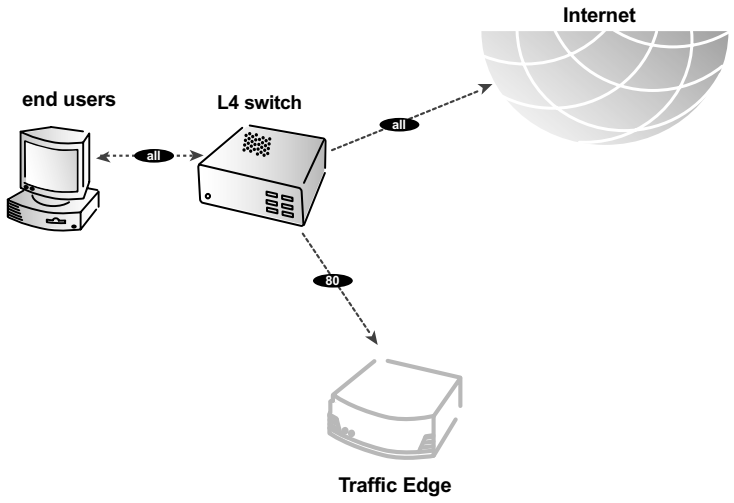


Figure 17 Using a Layer-4 switch to filter HTTP requests

Layer-4 switches offer the following features, depending on the particular switch:

- A Layer-4 switch that can sense downed hosts on the network and redirect traffic adds reliability.
- If a single Layer-4 switch feeds several Traffic Edge nodes, the switch handles load balancing among the nodes. Different switches might use different load balancing methods, such as round-robin or hashing. If a node becomes unavailable, the switch automatically redistributes the load. When the node returns to service, some switches automatically return the node to its previous workload, so that the node cache need not be repopulated; this feature is called *cache affinity*.

Note Inktomi recommends that you do *not* enable Traffic Edge virtual IP failover when a switch is providing load balancing in a cluster configuration. Also, if the switch provides a load balancing option that has cache affinity, Inktomi recommends that you use management-only clustering mode.

Using a WCCP-Enabled Router

Traffic Edge supports WCCP 1.0 and WCCP 2.0.

A WCCP 1.0-enabled router can send all port 80 (HTTP) traffic to Traffic Edge, as shown in *Figure 18*. The Traffic Edge ARM readdresses port 80 to the Traffic Edge proxy port (by default, port 8080). Traffic Edge processes the request as usual, retrieving the requested document from the cache if it is a hit and sending the response back to the client. Along the way, the ARM readdresses the proxy port in the response header to port 80 (undoing the readdressing it did on the way to Traffic Edge). The user then sees the response exactly as if it had been sent directly from the origin server.

A WCCP 2.0-enabled router works in the same way as a WCCP 1.0-enabled router. In addition to HTTP, WCCP 2.0 supports FTP, RTSP, PNA, MMS, and DNS.

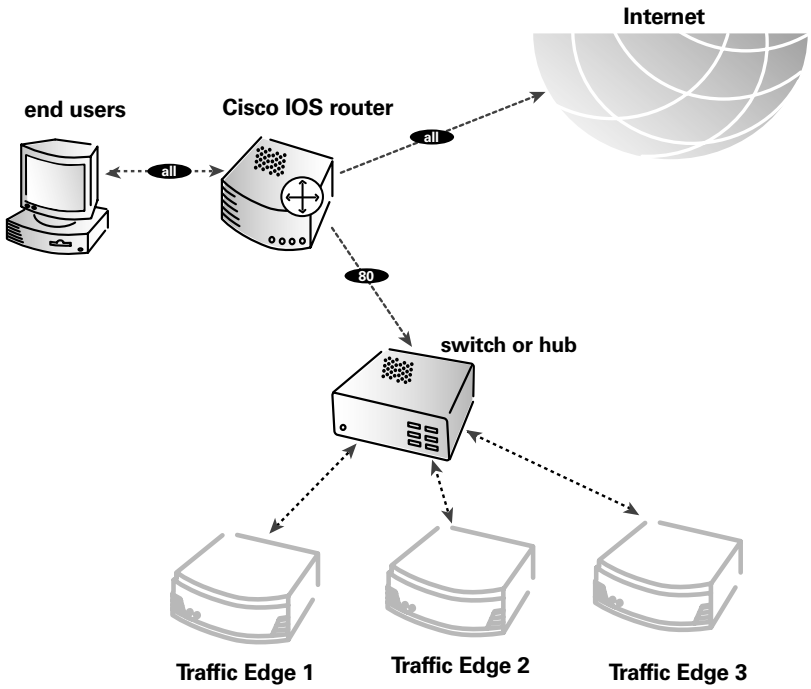


Figure 18 Using a Cisco IOS router to send port 80 traffic to several Traffic Edge nodes

WCCP provides the following routing benefits:

- The WCCP-enabled router and Traffic Edge exchange heartbeat messages, letting each other know they are running. The WCCP router automatically reroutes traffic if the Traffic Edge node becomes unavailable.
- If several Traffic Edge nodes receive traffic from a WCCP router, WCCP balances the load among the nodes. Such a group of Traffic Edge nodes is called a *WCCP cache farm*; refer to *WCCP Load Balancing, on page 116*.
- Traffic Edge handles node failure in WCCP cache farms. If one node becomes unavailable, its load is redistributed among the remaining nodes.
- In WCCP 2.0, you can use multiple routers. Traffic flowing through multiple routers can share the same pool of caches.

Enabling WCCP on Traffic Edge

Typically, you enable WCCP when you install Traffic Edge (refer to the *Traffic Edge Installation Guide* for installation instructions). However, you can enable WCCP on Traffic Edge at any time after installation either by setting options in Traffic Manager or by editing configuration files manually.

IMPORTANT Before you enable WCCP, make sure that your configuration meets the following requirements:

- The WCCP router is running the correct Cisco IOS release; refer to the *Traffic Edge Installation Guide* for information.
- If you are using several Traffic Edge nodes, determine whether you want the Traffic Edge nodes to use management-only clustering or full-clustering mode; refer to *WCCP Load Balancing, on page 116*.
- Each Traffic Edge must have the ARM enabled; refer to *Enabling the ARM Option, on page 107*.
- WCCP must be enabled on the router that is sending traffic to Traffic Edge. Instructions for enabling WCCP on Cisco routers is provided on the Cisco Systems website.

The following procedures describe how to enable WCCP on Traffic Edge after installation. Different procedures are provided for WCCP 1.0 and WCCP 2.0. Follow the procedure appropriate for your environment.

▼ To enable WCCP 1.0 from Traffic Manager:

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. Click the **Configure** tab.

   The **General** tab of the **My Proxy**/**Basic** button displays.
3. In the **Features** table, click the **ARM On** button (if it is not selected) and the **WCCP On** button in the **Networking** section.
4. Click the **Apply** button.
5. On the **Configure** tab, click the **Networking** button and then click the **WCCP** button.
6. Enable the **WCCP v1.0** option on the **General** tab and then click the **Apply** button.
7. Click the **WCCP v1.0** tab.
8. In the **WCCP Network Interface** field, enter the interface on the Traffic Edge system that receives traffic from the router.
9. In the **WCCP Router IP Address** field, enter the IP address of the router that sends traffic to Traffic Edge.
10. Click the **Apply** button.
11. Restart Traffic Edge by clicking the **Restart** button on the **Configure**/**My Proxy**/**Basic/General** tab.

▼ **To enable WCCP 1.0 manually:**

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Edit the following variables:

Variable	Description
proxy.config.arm.enabled	Make sure this variable is set to 1 so that the ARM option is enabled.
proxy.config.wccp.enabled	Set this variable to 1 to enable WCCP.
proxy.config.wccp.version	Set this variable to 1 to specify WCCP Version 1.0.
proxy.config.wccp.router_ip	Set this variable to specify the IP address of the WCCP router that is sending traffic to Traffic Edge.
proxy.config.wccp.ethernet_interface	Set this variable to specify the interface on the Traffic Edge system that receives traffic from the router.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -L` to restart Traffic Edge on the local node or `traffic_line -M` to restart Traffic Edge on all the nodes in a cluster.

▼ **To enable WCCP 2.0 from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. Click the **Configure** tab.

   The **General** tab of the **My Proxy**/**Basic** button displays.
3. In the **Features** table, click the **ARM On** button (if it is not selected) and the **WCCP On** button in the **Networking** section.
4. Click the **Apply** button.
5. On the **Configure** tab, click the **Networking** button and then click the **WCCP** button.
6. Enable the **WCCP v2.0** option on the **General** tab and then click the **Apply** button.
7. Click the **WCCP v2.0** tab.

Optional

8. In the **Security** area, click **Enabled** if you want Traffic Edge and your routers to authenticate each other.

Optional

9. In the **Password** field, enter the password used for service group authentication with the router. The password must match the authentication password for the service group specified on the router and must be a maximum of eight characters long; refer to *Using WCCP 2.0 Security, on page 114*.
10. Click the **Apply** button.
11. The **Configuration** section contains the `wccp_config.xml` file. Edit this file to configure service groups, router information, and multicast information; refer to *wccp_config.xml, on page 460*.
12. In the **Miscellaneous** section, enable the **Encapsulation** option to configure Traffic Edge to send encapsulated returned (bypassed) packets to the router.

13 Click the **Apply** button.

14 Restart Traffic Edge by clicking the **Restart** button on the **Configure**/**My Proxy**/**Basic/General** tab.

▼ To enable WCCP 2.0 manually:

1 In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.

2 Edit the following variables:

Variable	Description
proxy.config.arm.enabled	Make sure this variable is set to 1 so that the ARM option is enabled.
proxy.config.wccp.enabled	Set this variable to 1 to enable WCCP.
proxy.config.wccp.version	Set this variable to 2 to specify WCCP Version 2.0.
proxy.config.wccp2.security_enabled	*Optional*. Set this variable to 1 to enable security so that Traffic Edge and your routers can authenticate each other.
proxy.config.wccp2.password	Set this variable to specify the service group authentication password. The password must match the authentication password used for the service group on the router and must be a maximum of eight characters long; refer to *Using WCCP 2.0 Security, on page 114*.
proxy.config.wccp2.rev_encapsulation	Set this variable to 1 to configure Traffic Edge to send encapsulated returned (bypassed) packets to the router.

3 Save and close the `records.config` file.

4 In a text editor, open the `wccp_config.xml` file located in the Traffic Edge `config` directory.

5 Edit the file to configure service groups, router information, and multicast information; refer to *wccp_config.xml, on page 460*.

6 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

7 Run the command `traffic_line -L` to restart Traffic Edge on the local node or `traffic_line -M` to restart Traffic Edge on all the nodes in a cluster.

To check that the router is sending traffic to Traffic Edge, examine the statistics on the Traffic Manager **Monitor** tab; for example, check that the **Objects Served** statistic in the **My Proxy/Summary** section increases.

Note It might take more than 30 seconds for the router to report that a new proxy caching server has joined a service group.

Using WCCP 2.0 Security

If you are running WCCP 2.0 on your routers, you can enable security on the Traffic Edge node so that the node and your routers can authenticate each other. You enable the security option and provide the authentication password for Traffic Edge as described in *Enabling WCCP on Traffic Edge, on page 111*.

The authentication password you specify must match the authentication password configured on the router for each service group being intercepted. The following procedure provides an example of how to set an authentication password for different service groups on a WCCP 2.0-enabled router.

▼ **To set an authentication password on a WCCP 2.0-enabled router:**

1. Telnet to the router and switch to Enable mode.
2. At the prompt, enter the following command to configure the router from the terminal:

   ```
   config t
   ```

3. At the prompt, enter the following command for each service group that the router intercepts:

   ```
   hostname(config)# ip wccp service_group password password
   ```

   hostname is the hostname of the router you are configuring, *service_group* is the service group ID: for example, 0 for HTTP, and *password* is the password you want to use to authenticate Traffic Edge. This password must match the password you specify in Traffic Edge configuration.

IMPORTANT Traffic Edge supports one password only. Therefore, you must use the same password for all service groups.

4. Exit and save the router configuration.

Using Multicast Mode

To configure Traffic Edge to run in multicast mode, you must first enable multicast mode and then specify the multicast IP address as described in *Enabling WCCP on Traffic Edge, on page 111*. In addition, you must set the multicast address on your routers for each service group being intercepted. The following procedure provides an example of how to set the multicast address for different service groups on a WCCP 2.0-enabled router.

▼ **To set the multicast address on a WCCP 2.0-enabled router:**

1. Telnet to the router and switch to Enable mode.
2. At the prompt, enter the following command to configure the router from the terminal:

   ```
   config t
   ```

3. At the prompt, enter the following command for each service group that the router intercepts:

   ```
   hostname(config)# ip wccp service_group group-address multicast_address
   ```

   hostname is the hostname of the router you are configuring, *service_group* is the service group ID: for example, 0 for HTTP, and *multicast_address* is the IP multicast address.

4 At the prompt, enter the following command to configure the network interface:

```
interface interface_name
```

interface_name is the network interface on the router that is being intercepted and redirected.

5 At the prompt, enter the following command for each service group that the router intercepts:

```
hostname(config-if)# ip wccp service_group group-listen
```

6 Exit and save the router configuration.

L2 Redirection

Traffic Edge supports L2 redirection, which enables a router to redirect packets to a Traffic Edge node on the same subnet by sending the packets to the MAC address instead of encapsulating the original IP packet meant for the origin server in another IP packet (GRE encapsulation). Traffic Edge supports L2 redirection in forward mode, to receive client packets that the router redirects, and in return mode, to return the client packets it received from the router back to the router.

To enable Traffic Edge L2 redirection support, you must edit the `records.config` file. Use the following procedure.

▼ **To enable L2 redirection support:**

1 In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.

2 Edit the following variable:

Variable	Description
proxy.config.wccp2.layer_2_rewrite.enabled	Set this variable to one of the following values: 0 to disable L2 redirection. This is the default value. 1 to use L2 redirection only when the router advertises L2 for *both* forward mode and return mode; for example, if the router advertises L2 for forward mode but GRE encapsulation for return mode, Traffic Edge uses GRE encapsulation for both forward and return mode. 2 to use L2 redirection for both forward and return mode even though the router advertises L2 redirection for forward mode but GRE encapsulation for return mode. Note: You cannot configure Traffic Edge to use L2 redirection for forward mode and GRE encapsulation for return mode.

3 Save and close the `records.config` file.

4 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

5 Run the command `traffic_line -L` to restart Traffic Edge on the local node or `traffic_line -M` to restart Traffic Edge on all the nodes in a cluster.

HTTP ARM Bypass and WCCP

If Traffic Edge has an ARM bypass rule (discussed in *Interception Bypass (HTTP), on page 120*), Traffic Edge forwards particular client requests directly to the origin server, bypassing the cache. Bypassed requests are unchanged by the ARM; they retain their client source IP addresses. In WCCP 1.0, ARM bypass rules cannot work if the WCCP router is also the Traffic Edge default gateway router. The WCCP router sends port 80 traffic to the Traffic Edge *and* it serves as the Traffic Edges' default gateway or *next hop* to the Internet. Bypassed requests go to the WCCP router, which sends them back to Traffic Edge.

In WCCP 2.0, you can exclude certain router interfaces from redirection. Traffic Edge bypass rules can work if you exclude the router interface on which Traffic Edge is connected from using WCCP. You can do this by setting the router configuration command `ip wccp redirect exclude in` (refer to the WCCP documentation from Cisco Systems for information about router configuration).

WCCP Load Balancing

If a WCCP router serves several nodes, the router balances load among the Traffic Edges. The router sends each node requests aimed at a particular range of IP addresses, so that each node is responsible for caching content residing at particular IP addresses. If a node becomes unavailable, its traffic is redistributed.

Traffic Edge also supports cache affinity. If a node becomes unavailable and then recovers, Traffic Edge returns the node to its former load distribution. Therefore, the node's cache does not need to be repopulated.

The WCCP cache farm acts as a simple form of distributed cache, which is sufficient for many applications. A WCCP-enabled network device distributes traffic to individual Traffic Edges based on the IP address of the origin server. Each node caches objects requested from a particular set of origin servers, which belong to that node's assigned range of destination IP addresses.

Traffic Edge full-clustering mode is not required for WCCP and you can run Traffic Edge nodes in management-only clustering mode. During Traffic Edge installation, if you select clustering and enable WCCP, management-only clustering is enabled by default. Management-only clustering conserves CPU resources and slightly improves performance over full clustering. Refer to *Chapter 8, Traffic Edge Clusters*, for details.

Busy origin servers are often mapped to several IP addresses (using a DNS round-robin mechanism). Using WCCP-based load balancing alone, each of these different IP addresses could be allotted to different Traffic Edge nodes. This can result in a slightly lower hit rate and wasted cache space, since the same content is being replicated across nodes. The Traffic Edge full clustering mode ensures that all requests to a specific page on that origin server (no matter which IP address is used) are cached on the same node.

With full clustering, objects are distributed among nodes according to their URLs; WCCP distributes objects according to destination IP address. If a particular IP address is receiving many requests, WCCP load balancing might lead to a hot spot, where all of that site's traffic is cached on one node, instead of being distributed among the nodes. The Traffic Edge full-clustering mode distributes different pages from the busy site to different Traffic Edge nodes.

In general, if load-handling capacity and latency are most important, Inktomi recommends management-only clustering in WCCP environments. If hit rate, bandwidth savings, and better load balancing are most important, then full clustering can provide an improvement in WCCP environments.

If you are running clustered Traffic Edges, Inktomi recommends that you do *not* enable virtual IP failover in WCCP environments. The Traffic Edge WCCP failover mechanism handles node failures and restarts; refer to *Using Virtual IP Failover, on page 155*.

Slow Start

When multiple Traffic Edge nodes receive WCCP packets from a router, the router distributes the load between them. Typically, the Traffic Edge node that is designated to be the *leader* (usually the one with the lowest IP address) determines how much traffic to distribute to each other Traffic Edge node. The leader gets information about the other nodes in the network from WCCP protocol packets that are exchanged every 10 seconds (not the packets that the router redirects to Traffic Edge). Initially, the leader thinks it is the only Traffic Edge node on the network and asks the router to send all of the traffic to it. When the leader discovers another Traffic Edge node, it distributes 50 percent of the load to the new node, and so on. When the leader gets more packets from the router, it updates the list of Traffic Edge nodes on the network (it might take 30 seconds or more for all the Traffic Edge nodes and routers involved to know about each other).

If you deploy more than one Traffic Edge node because you know that one node alone cannot handle all the traffic, the first Traffic Edge node initially sees 100 percent of the traffic, which is greater than it is designed for, and the second node might see 50 percent of the traffic, which might also be greater than it is designed for. With the Slow Start option, you can configure Traffic Edge to redirect only a certain percentage (10 percent by default) of the traffic from the router to the leader (the rest is bypassed and goes to the origin servers). After a certain period of time (15 seconds by default) the router redirects 20 percent of the traffic (if there are two Traffic Edge nodes known to the leader by this time, each will get 10 percent). The leader continues until 100 percent of the traffic is distributed.

To configure the Slow Start option, you must edit the `records.config` file. Use the following procedure.

▼ To configure Slow Start:

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Edit the following variables:

Variable	Description
proxy.config.wccp2.slow_start.enabled	Set this variable 1 to enable the Slow Start option. The default value is 0 (disabled).

Variable	Description
proxy.config.wccp2.slow_start.increment	Set this variable to specify the percentage of traffic that the router sends to the leader Traffic Edge at every interval specified by the proxy.config.wccp2.slow_start.interval variable. The default value is 10. Note: Setting this value to 100 percent, disables the Slow Start option.
proxy.config.wccp2.slow_start.interval	Set this variable to specify the number of seconds after which the leader increments the percentage of traffic. The default value is 15.

3 Save and close the `records.config` file.

4 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

5 Run the command `traffic_line -L` to restart Traffic Edge on the local node or `traffic_line -M` to restart Traffic Edge on all the nodes in a cluster.

Using Policy-Based Routing

Instead of the WCCP protocol, you can use the policy routing capabilities of a router to send traffic to Traffic Edge. WCCP or a Layer-4 switch are generally preferable to this configuration because policy-based routing has a performance impact on the router, and policy-based routing does not support load balancing or heartbeat messaging.

Figure 19 illustrates policy routing for HTTP requests.

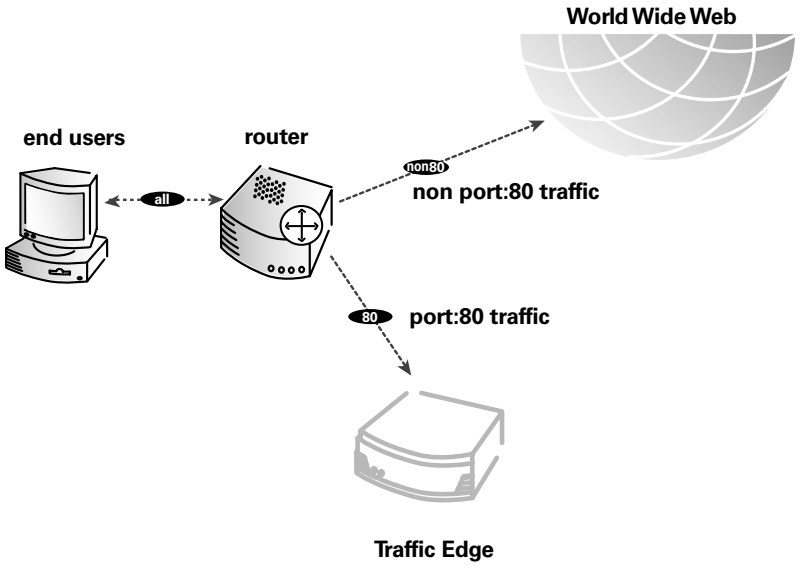


Figure 19 Using a router to filter HTTP requests

Figure 19 demonstrates the following steps:

1. All client Internet traffic is sent to a router that feeds Traffic Edge.
2. The router sends port 80 (HTTP) traffic to Traffic Edge and sends the remaining traffic to the next hop router.
3. The ARM translates intercepted requests into Traffic Edge requests.
4. Translated requests are sent to Traffic Edge.

Web objects to be served transparently are readdressed by the ARM on the return path to the client, so that the documents appear to have come straight from the origin server. A Traffic Edge cluster with virtual IP failover adds reliability; if one node fails, another node can take up its transparency requests; refer to *Using Virtual IP Failover, on page 155*.

Using Software-Based Transparency Solutions

You can deploy Traffic Edge transparently without adding routers or switches by using routing software on the Traffic Edge node. In this case, Traffic Edge is a software router and directs all traffic through the Traffic Edge machine. This solution can be useful in low-traffic situations, where the performance cost of using the Traffic Edge machine as a router is not high.

On UNIX systems, you can use the `routed` and `gated` daemons as a software-based routing solution. The `routed` daemon is a bundled part of all normal UNIX distributions. The `gated` daemon is an extensible commercial software package from the Merit GateD Consortium.

When you use routing software on Traffic Edge:

- All Internet traffic goes through Traffic Edge from machines behind it in the network.
- The routing software routes all nontransparent requests to the Internet; it routes port 80 HTTP requests to Traffic Edge.
- The ARM translates intercepted requests into Traffic Edge requests.
- Translated requests are sent to Traffic Edge.
- Web objects to be served transparently are readdressed by the ARM on the return path to the client, so that the objects appear to have come straight from the origin server.

Although Traffic Edge machines can function as routers, they are not expressly designed to be routers. For reliability, you can use a Traffic Edge cluster with the virtual IP failover option. If one node fails, another cluster node takes over. The Traffic Edge cluster failover mechanism is similar to the Hot Standby Router Protocol (HSRP).

Interception Bypass (HTTP)

A very small number of clients and servers do not work correctly with web proxies. Some of the reasons include:

- Client software bugs (customized, noncommercial browsers).
- Server software bugs.
- Applications that send nonHTTP traffic over HTTP ports as a way of defeating security restrictions.
- Server IP authentication (the origin server limits access to a few client IP addresses, but the Traffic Edge IP address is different, so it cannot get access). This is not in frequent use because many ISPs dynamically allocate client IP dial-up addresses, and more secure cryptographic protocols are now more often used.

Web proxies are very common in corporate and Internet use, so the frequency of interoperability problems is extremely rare. However, Traffic Edge contains an adaptive learning module that recognizes interoperability problems caused by transparent proxy caching and automatically bypasses the traffic around Traffic Edge without operator intervention.

Traffic Edge follows two types of bypass rules:

- *Dynamic* (also called *adaptive*) bypass rules are generated dynamically if you configure Traffic Edge to bypass the cache when it detects nonHTTP traffic on port 80 or when it encounters certain HTTP errors; refer to *Using Dynamic Bypass Rules*, below.
- *Static* bypass rules must be manually configured in the `bypass.config` file; refer to *Using Static Bypass Rules, on page 125*.

Do not confuse bypass rules with client access control lists. Bypass rules are generated in response to interoperability problems. Client access control is simply restriction of the client IP addresses that can access the Traffic Edge cache, as described in *Controlling Client Access to the Proxy Cache, on page 203*.

Using Dynamic Bypass Rules

When configured to do so, Traffic Edge can detect certain protocol interoperability errors. As it detects errors, it configures the ARM to bypass the proxy for those clients and/or servers causing the errors.

In this way, the very small number of clients or servers that do not operate correctly through proxies are autodetected and routed around Traffic Edge so that they can continue to function normally (but without the improvement of caching).

You can configure Traffic Edge to dynamically bypass the cache for any of the following errors.

Error Code	Description
N/A	NonHTTP traffic on port 80
400	Bad Request
401	Unauthorized
403	Forbidden (authentication failed)
405	Method Not Allowed

Error Code	Description
406	Not Acceptable (access)
408	Request Timeout
500	Internal Server Error

For example, when Traffic Edge is configured to bypass on authentication failure (`403 Forbidden`), if any request to an origin server returns a 403 error, Traffic Edge generates a destination bypass rule for the origin server's IP address. All requests to that origin server are bypassed until you restart Traffic Edge.

In another example, if the ARM detects that a client is sending a nonHTTP request on port 80 to a particular origin server, Traffic Edge generates a source/destination rule. All requests from that particular client to the origin server are bypassed; requests from other clients are not bypassed.

Bypass rules that are generated dynamically are purged after a Traffic Edge restart. If you want to preserve dynamically generated rules, you can save a snapshot of the current set of bypass rules; refer to *Viewing the Current Set of Bypass Rules, on page 125*.

To prevent Traffic Edge from bypassing certain IP addresses dynamically, you can set dynamic *deny* bypass rules in the `bypass.config` file. Deny bypass rules can prevent Traffic Edge from bypassing itself. For information about setting dynamic deny bypass rules, refer to *bypass.config, on page 367*.

Setting Dynamic Bypass Rules

By default, Traffic Edge is not configured to bypass the cache when it encounters HTTP errors or nonHTTP traffic on port 80. You must enable dynamic bypass rules either by setting options in Traffic Manager or by editing the `records.config` file manually. Both procedures are provided below.

▼ **To set dynamic bypass rules from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. Click the **Configure** tab.

   The **General** tab of the **My Proxy**/**Basic** button displays.
3. In the **Features** table, make sure that the **ARM On** button is enabled in the **Networking** section.

Note If you enable the **ARM On** button, you must click the **Apply** button to save the change.

4. On the **Configure** tab, click the **Networking** button and then click the **ARM** button.
5. Click the **Dynamic Bypass** tab.
6. Enable the **Dynamic Bypass** option.
7. In the **Behavior** section, select the dynamic bypass rules you want to use.
8. Click the **Apply** button.
9. Restart Traffic Edge by clicking the **Restart** button on the **Configure**/**My Proxy**/**Basic/General** tab.

▼ **To set dynamic bypass rules manually:**

1. In a text editor, open the `records.config` file located in the `config` directory.
2. Edit the following variables in the `ARM (Transparency Configuration)` section of the file:

Variable	Description
proxy.config.arm.bypass_dynamic_enabled	Set this variable to 1 to enable dynamic bypass.
proxy.config.arm.bypass_use_and_rules_bad_client_request	Set this variable to 1 to enable dynamic source/destination bypass when there is nonHTTP traffic on port 80.
proxy.config.arm.bypass_use_and_rules_400	Set this variable to 1 to enable dynamic source/destination bypass when an origin server returns a 400 error.
proxy.config.arm.bypass_use_and_rules_401	Set this variable to 1 to enable dynamic source/destination bypass when an origin server returns a 401 error.
proxy.config.arm.bypass_use_and_rules_403	Set this variable to 1 to enable dynamic source/destination bypass when an origin server returns a 403 error.
proxy.config.arm.bypass_use_and_rules_405	Set this variable to 1 to enable dynamic source/destination bypass when an origin server returns a 405 error.
proxy.config.arm.bypass_use_and_rules_406	Set this variable to 1 to enable dynamic source/destination bypass when an origin server returns a 406 error.
proxy.config.arm.bypass_use_and_rules_408	Set this variable to 1 to enable dynamic source/destination bypass when an origin server returns a 408 error.
proxy.config.arm.bypass_use_and_rules_500	Set this variable to 1 to enable dynamic source/destination bypass when an origin server returns a 500 error.
proxy.config.arm.bypass_on_bad_client_request	Set this variable to 1 to enable dynamic destination bypass when there is nonHTTP traffic on port 80.
proxy.config.arm.bypass_on_400	Set this variable to 1 to enable dynamic destination bypass when an origin server returns a 400 error.
proxy.config.arm.bypass_on_401	Set this variable to 1 to enable dynamic destination bypass when an origin server returns a 401 error.

Variable	Description
proxy.config.arm.bypass_on_403	Set this variable to 1 to enable dynamic destination bypass when an origin server returns a 403 error.
proxy.config.arm.bypass_on_405	Set this variable to 1 to enable dynamic destination bypass when an origin server returns a 405 error.
proxy.config.arm.bypass_on_406	Set this variable to 1 to enable dynamic destination bypass when an origin server returns a 406 error.
proxy.config.arm.bypass_on_408	Set this variable to 1 to enable dynamic destination bypass when an origin server returns a 408 error.
proxy.config.arm.bypass_on_500	Set this variable to 1 to enable dynamic destination bypass when an origin server returns a 500 error.

IMPORTANT For a dynamic source/destination bypass rule to work, you must also enable the equivalent destination bypass rule; for example, when you set the variable `proxy.config.arm.bypass_use_and_rules_403` to `1`, you must also set the variable `proxy.config.arm.bypass_on_403` to `1`.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -L` to restart Traffic Edge on the local node or `traffic_line -M` to restart Traffic Edge on all the nodes in a cluster.

Viewing Dynamic Bypass Statistics

Traffic Edge tallies bypassed requests for each type of dynamic bypass trigger; for example, Traffic Edge counts all requests that are bypassed in response to a 401 error.

You can view the dynamic bypass statistics either from Traffic Manager or from Traffic Line.

▼ **To view dynamic bypass statistics from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Monitor** tab, click the **Networking** button and then click the **ARM** button.

   The statistics display in the **HTTP Bypass Statistics** section of the table.

▼ **To view dynamic bypass statistics from Traffic Line:**

1 In UNIX, log on to a Traffic Edge node as the Traffic Edge administrator and navigate to the Traffic Edge `bin` directory.

In Windows, open a Command Prompt window and navigate to the Traffic Edge `bin` directory.

2 Enter the following command and then press Return:

`traffic_line -r` *`variable`*

`variable` is one of the following statistics:

Variable	Description
proxy.process.arm.num_bypass_on_bad_client_request	Displays the number of times Traffic Edge bypassed the cache because it detected nonHTTP traffic on port 80.
proxy.process.arm.num_bypass_on_400	Displays the number of times Traffic Edge bypassed the cache because it detected an HTTP 400 error.
proxy.process.arm.num_bypass_on_401	Displays the number of times Traffic Edge bypassed the cache because it detected an HTTP 401 error.
proxy.process.arm.num_bypass_on_403	Displays the number of times Traffic Edge bypassed the cache because it detected an HTTP 403 error.
proxy.process.arm.num_bypass_on_405	Displays the number of times Traffic Edge bypassed the cache because it detected an HTTP 405 error.
proxy.process.arm.num_bypass_on_406	Displays the number of times Traffic Edge bypassed the cache because it detected an HTTP 406 error.
proxy.process.arm.num_bypass_on_408	Displays the number of times Traffic Edge bypassed the cache because it detected an HTTP 408 error.
proxy.process.arm.num_bypass_on_500	Displays the number of times Traffic Edge bypassed the cache because it detected an HTTP 500 error.

Using Static Bypass Rules

In addition to adaptively learning what to bypass, Traffic Edge allows you to manually configure bypass rules to direct requests from certain clients or to particular origin servers around Traffic Edge. Unlike dynamic bypass rules that are purged when you restart Traffic Edge, these static bypass rules are saved in a configuration file.

For example, you might want client IP addresses that did not pay for a caching service to be steered around the cache, while paying clients can obtain the benefits of caching. Or you might want to remove some servers from caching lists because they do not want to have their pages cached.

You can configure three types of static bypass rules:

- *Source bypass*, with which Traffic Edge bypasses a particular source IP address or range of IP addresses; for example, you can use this solution to bypass clients who want to opt out of a caching solution.
- *Destination bypass*, with which Traffic Edge bypasses a particular destination IP address or range of IP addresses; for example, these could be origin servers that use IP authentication based on the client's real IP address. Destination bypass rules prevent Traffic Edge from caching an entire site. You will experience hit rate impacts if the site you bypass is popular.
- *Source/destination pair bypass*, with which Traffic Edge bypasses requests that originate from the specified source to the specified destination; for example, you could route around specific client-server pairs that experience broken IP authentication or out of band HTTP traffic problems when cached.

  Source/destination bypass rules might be preferable to destination rules because they block a destination server only for those particular users that experience problems.

To configure static bypass rules, edit the `bypass.config` file; refer to *bypass.config, on page 367*.

Viewing the Current Set of Bypass Rules

The ARM has a supporting utility called `print_bypass` that allows you to view the current dynamic and static bypass rules.

▼ **To view all current dynamic and static bypass rules:**

1. In UNIX, log on to a Traffic Edge node as the Traffic Edge administrator and navigate to the Traffic Edge `bin` directory.

   In Windows, open a Command Prompt window and navigate to the Traffic Edge `bin` directory.

2. Enter the following command at the prompt and press Return:

   ```
   print_bypass
   ```

   All current static and dynamic bypass rules display onscreen. The rules are sorted by IP address. You can direct the output of `print_bypass` to a file and save it.

Configuring ARM Security

To prevent unauthorized access to machines running Traffic Edge, you can configure the ARM to utilize an access control list employing administrator-specified rules to either allow or deny other computers from communicating with the machine. This enables you to create a firewall in front of Traffic Edge to deny potentially malicious packets from even reaching the TCP/IP stack on the machine. Refer to *Controlling Host Access to the Traffic Edge Machine, on page 204*.

Connection Load Shedding (HTTP and FTP)

The load shedding feature prevents client request overloads. When there are more client connections than the specified limit, the ARM forwards incoming requests directly to the origin server. The default client connection limit is 1 million connections. You can set the limit to better suit your needs either by using Traffic Manager or by editing a configuration file manually. Both procedures are described below.

▼ To set the client connection limit from Traffic Manager:

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Networking** button and then click the **Connection Management** button.
3. Click the **Load Shedding** tab.
4. In the **Maximum Connections** field, specify the maximum number of client connections allowed before the Traffic Edge ARM starts forwarding requests directly to the origin server.
5. Click the **Apply** button.
6. Restart Traffic Edge by clicking the **Restart** button on the **Configure**/**My Proxy**/**Basic/General** tab.

▼ To set the client connection limit manually:

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Edit the following variable:

Variable	Description
proxy.config.arm.loadshedding.max_connections	Set this variable to specify the maximum number of client connections allowed before the Traffic Edge ARM starts forwarding requests directly to the origin server.

3. Save and close the `records.config` file.
4. Restart Traffic Edge.

Reducing DNS Lookups

If you are running Traffic Edge in transparent proxy mode, you can enable the *Always Query Destination* option to reduce the number of DNS lookups and improve Traffic Edge response time. When enabled, the Always Query Destination option configures Traffic Edge to always obtain the original destination IP address of incoming requests from the ARM driver. Traffic Edge then uses that IP address to determine the origin server instead of doing a DNS lookup on the hostname of the request. Because the client already performed a DNS lookup, Traffic Edge does not have to.

Note Inktomi recommends that you do not enable the *Always Query Destination* option if Traffic Edge is running in *both* explicit proxy caching mode and transparent proxy caching mode. Refer to *How do you configure Traffic Edge to serve only transparent requests?, on page 478*, for information about running Traffic Edge in transparent proxy caching mode only. In explicit proxy caching mode, the client does not perform a DNS lookup on the hostname of the origin server, so Traffic Edge must perform a DNS lookup.

▼ **To enable the Always Query Destination option:**

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Edit the following variable:

Variable	Description
proxy.config.arm.always_query_dest	Set this variable to 1 to enable the Always Query Destination option.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

IP Spoofing (HTTP)

The IP spoofing option configures Traffic Edge to use the IP address of the client when communicating with origin servers instead of its own IP address. As a result, HTTP requests appear to come from the client rather than from Traffic Edge.

Before enabling the IP spoofing option, you must enable the ARM. The Traffic Edge ARM translates the source IP address of the outgoing request and the IP address of the incoming response.

When you enable the IP spoofing option, Traffic Edge uses the client IP address instead of its own in *all* origin server communications. This option affects nontransparent requests as well as transparent requests.

IMPORTANT Asymmetric routing is necessary for the IP spoofing option to work. Make sure that all traffic from the origin server to the client goes through Traffic Edge.

IMPORTANT When you enable the IP spoofing option, Traffic Edge disables the re-use of server sessions, which might affect Traffic Edge performance.

IMPORTANT Do not use the IP spoofing option if you are using a WCCP-enabled router.

You can enable the IP spoofing option either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ To enable the IP spoofing option from Traffic Manager:

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. Click the **Configure** tab.

   The **General** tab of the **My Proxy**/**Basic** button displays.
3. In the **Features** table on the **General** tab, click the **ARM On** button in the **Networking** section.
4. Click the **Apply** button.
5. On the **Configure** tab, click the **Networking** button and then click the **ARM** button.
6. Enable **IP Spoofing** on the **General** tab.
7. Click the **Apply** button.
8. Restart Traffic Edge by clicking the **Restart** button on the **Configure**/**My Proxy**/**Basic/General** tab.

▼ To enable the IP spoofing option manually:

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Edit the following variables:

Variable	Description
proxy.config.arm.enabled	Set this variable to 1 to enable the ARM.
proxy.config.http.outgoing_ip_spoofing_enabled	Set this variable to 1 to enable IP spoofing.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -L` to restart Traffic Edge on the local node or `traffic_line -M` to restart Traffic Edge on all the nodes in a cluster.

CHAPTER 7 Reverse Proxy and HTTP Redirects

As a reverse proxy cache, Traffic Edge serves requests on behalf of origin servers. Traffic Edge is configured in such a way that it appears to clients like a normal origin server.

This chapter discusses the following topics:

Understanding Reverse Proxy Caching

In forward proxy caching, Traffic Edge handles web requests to distant origin servers on behalf of the clients requesting the content. *Reverse proxy caching* (also known as *server acceleration* or *virtual web hosting*) is different in that Traffic Edge acts as a proxy cache on behalf of the origin servers that store the content. Traffic Edge is configured to be *the* origin server the user is trying to connect to (typically, the advertised hostname of the origin server resolves to Traffic Edge, which is acting as the real origin server).

Reverse Proxy Solutions

There are many ways in which Traffic Edge can be used as a reverse proxy. Here are a few example scenarios.

You can use Traffic Edge in reverse proxy mode to:

- Offload heavily used origin servers
- Deliver content efficiently in geographically dispersed areas
- Provide security for origin servers that contain sensitive information

Offloading Heavily Used Origin Servers

Traffic Edge can absorb the main origin server request traffic to improve the speed and quality of web serving by reducing load and hot spots on backup origin servers; for example, a web hoster can maintain a scalable Traffic Edge serving engine and a set of low-cost, low-performance, less-reliable PC origin servers as backup servers. In fact, a

single Traffic Edge can act as the virtual origin server for multiple backup origin servers, as shown in *Figure 20*.

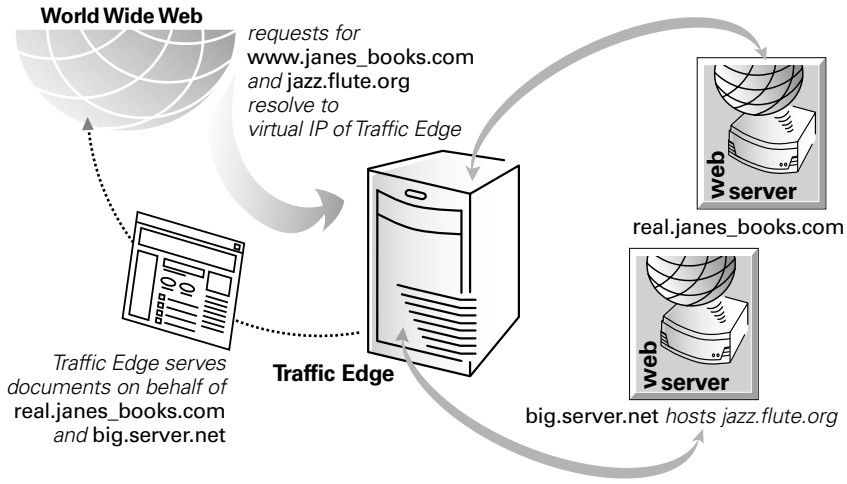


Figure 20 Traffic Edge as reverse proxy for a pair of origin servers

Delivering Content in Geographically Dispersed Areas

Traffic Edge can be used in reverse proxy mode to accelerate origin servers that provide content to geographically dispersed areas. Caches can be easier to manage and more cost-effective than replicating data. For example, Traffic Edge can be used as a mirror site on the far side of a trans-Atlantic link to serve users without having to fetch the request and content across expensive international connections. Unlike replication, where hardware must be configured to replicate all data and to handle peak capacity, Traffic Edge dynamically adjusts to best utilize the serving and storing capacity of the hardware. Also, Traffic Edge is designed to keep content fresh automatically, therefore eliminating the complexity of updating remote origin servers.

Providing Security for an Origin Server

Traffic Edge can be used in reverse proxy mode to provide security for an origin server. If an origin server contains sensitive information that you want to keep secure inside your firewall, you can use a Traffic Edge outside the firewall as a reverse proxy for that origin server. When outside clients try to access the origin server, their requests go to Traffic Edge instead. If the desired content is *not* sensitive, it can be served from the cache. If the content *is* sensitive and not cacheable, Traffic Edge obtains the content from the origin server (the firewall allows only Traffic Edge access to the origin server). The sensitive content resides on the origin server, safely inside the firewall.

How Does Reverse Proxy Work?

When a browser makes a request, it normally sends that request directly to the origin server. When Traffic Edge is in reverse proxy mode, it must intercept the request for that origin server.

Typically, this is done by setting up the DNS entry for the origin server (the origin server's *advertised* hostname) to resolve to the Traffic Edge IP address. When Traffic Edge is

configured as the origin server, the browser will connect to Traffic Edge rather than the origin server.

Note The origin server's hostname and its advertised hostname cannot be the same or there would be a DNS conflict.

The way that Traffic Edge receives and processes requests for content in reverse proxy mode differs according to protocol:

- For information about using and configuring reverse proxy for HTTP requests, refer to *HTTP Reverse Proxy, on page 131*.
- For information about using and configuring reverse proxy for FTP requests, refer to *FTP Reverse Proxy, on page 138*.
- For information about using and configuring reverse proxy for streaming media requests, refer to *Streaming Media Reverse Proxy, on page 142*.

HTTP Reverse Proxy

In reverse proxy mode, Traffic Edge serves HTTP requests on behalf of a web server (typically, the web server hostname resolves to the Traffic Edge IP address).

Figure 21 illustrates how Traffic Edge in reverse proxy mode serves an HTTP request from a client browser.

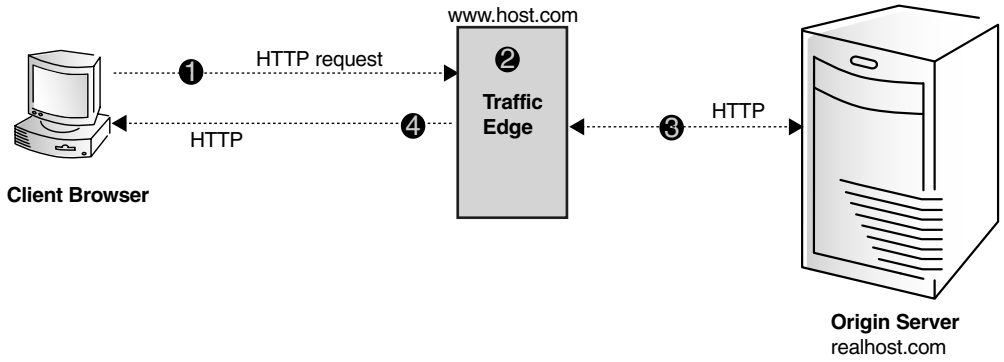


Figure 21 HTTP reverse proxy

Figure 21 demonstrates the following steps:

1. A client browser sends an HTTP request addressed to a host called `www.host.com` on port 80. Traffic Edge receives the request because it is acting as the origin server (the origin server's advertised hostname resolves to Traffic Edge).
2. Traffic Edge locates a map rule in the `remap.config` file and remaps the request to the specified origin server (`realhost.com`).
3. Traffic Edge opens an HTTP connection to the origin server.
4. If the request is a cache hit and the content is fresh, Traffic Edge sends the requested object to the client from the cache; if not, Traffic Edge obtains the requested object from the origin server, sends the object to the client and saves a copy in its cache.

To configure HTTP reverse proxy, you must perform the following tasks:

- Create mapping rules in the `remap.config` file; refer to *Creating Mapping Rules for HTTP Requests*, below.
- Enable the reverse proxy option; refer to *Enabling HTTP Reverse Proxy, on page 134*.

In addition to the tasks listed above, you can set optional reverse proxy options; refer to *Setting Optional HTTP Reverse Proxy Options, on page 135*.

Creating Mapping Rules for HTTP Requests

In forward proxy caching, Traffic Edge acts as a proxy server and receives proxy requests. In reverse proxy caching, Traffic Edge needs to act as an origin server rather than a proxy server, meaning that it receives server requests, not proxy requests. To satisfy proxy requests, Traffic Edge must construct a proxy request from the server request.

In HTTP, server requests differ from proxy requests in that server requests do not specify the entire URL, just the path. A server request might look like this:

```
GET /index.html HTTP/1.0
Host: real.janes_books.com
```

However, the corresponding proxy request would look like this:

```
GET http://real.janes_books.com/index.html HTTP/1.0
Host: real.janes_books.com
```

Traffic Edge can construct a proxy request from a server request by using the server information in the host header.

However, the correct proxy request must contain the hostname of the origin server, not the advertised hostname that the name servers associate to Traffic Edge. The advertised hostname is the name that appears in the host header; for example, for the origin server `real.janes_books.com` in *Figure 20, on page 130*, the server request and host header would be:

```
GET /index.html HTTP/1.0
Host: www.janes_books.com
```

And the correct proxy request should be

```
GET http://real.janes_books.com/index.html HTTP/1.0
Host: real.janes_books.co
```

To translate `www.janes_books.com` to `real.janes_books.com`, Traffic Edge needs a set of URL rewriting rules (*mapping rules*). Mapping rules are described in *Using Mapping Rules for HTTP Requests, on page 133*.

Generally, you use reverse proxy mode to support more than one origin server. In this case, all of the advertised hostnames resolve to the IP address or virtual IP address of Traffic Edge. Using host headers, Traffic Edge is able to translate server requests for any number of servers into proxy requests for those servers.

If Traffic Edge receives requests from older browsers that do not support host headers, Traffic Edge can route these requests directly to a specific server or send the browser to a URL containing information about the problem; refer to *Setting Optional HTTP Reverse Proxy Options, on page 135*.

Handling Origin Server Redirect Responses

Origin servers often send redirect responses (redirects) back to browsers, redirecting them to different pages.; for example, if an origin server is overloaded, it might redirect browsers to a less loaded server. Origin servers also redirect when web pages have moved to different locations. When Traffic Edge is configured as a reverse proxy, it must readdress redirects from origin servers so that browsers are redirected to Traffic Edge, not to another origin server.

To readdress redirects, Traffic Edge uses reverse-map rules. In general, you should set up a reverse-map rule for each map rule. To create reverse-map rules, refer to *Using Mapping Rules for HTTP Requests*, below.

Using Mapping Rules for HTTP Requests

Traffic Edge uses two types of mapping rules for HTTP reverse proxy:

- A *map rule* translates the URL in client requests into the URL where the content is located. When Traffic Edge in reverse proxy mode receives an HTTP client request, it first constructs a complete request URL from the relative URL and its headers. Traffic Edge then compares the complete request URL with its list of target URLs in the `remap.config` file, looking for a match. For the request URL to match a target URL, the following conditions must be true:
  - The scheme of both URLs must be the same
  - The host in both URLs must be the same (if the request URL contains an unqualified hostname, it will never match a target URL with a fully qualified hostname)
  - The ports in both URLs must be the same (if no port is specified in a URL, the default port for the scheme of the URL is used)
  - The path portion of the target URL must match a prefix of the request URL path

  If Traffic Edge finds a match, it translates the request URL into the replacement URL listed in the map rule. It sets the host and path of the request URL to match the replacement URL. If the URL contains path prefixes, Traffic Edge removes the prefix of the path that matches the target URL path and substitutes it with the path from the replacement URL.

  If two mappings match a request URL, Traffic Edge applies the first mapping listed in the `remap.config` file.

- A *reverse-map rule* translates the URL in origin server redirect responses to point to the Traffic Edge so that clients are redirected to Traffic Edge instead of accessing an origin server directly; for example, if there is a directory `/pub` on an origin server at `www.molasses.com` and a client sends a request to that origin server for `/pub`, the origin server might reply with a redirect to `http://www.test.com/pub/` to let the client know that it was a directory it had requested, not a document. (A common use of redirects is to normalize URLs so that clients can bookmark documents properly.)

  Traffic Edge uses reverse-map rules to prevent redirects from origin servers from causing clients to bypass Traffic Edge in favor of direct access to the origin servers.

Both map and reverse-map rules consist of a *target* (origin) URL and a *replacement* (destination) URL. In a *map* rule, the target URL points to Traffic Edge and the replacement URL specifies where the original content is located. In a *reverse-map* rule, the target URL specifies where the original content is located and the replacement URL points

to Traffic Edge. Traffic Edge stores mapping rules in the `remap.config` file located in the Traffic Edge `config` directory.

You can create mapping rules either by using Traffic Manager or by editing a configuration file manually.

▼ **To create mapping rules from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Content Routing** button and then click the **Mapping and Redirection** button.
3. In the **URL Mapping Rules** section of the **General** tab, click the **Edit File** button.

   The configuration file editor for the `remap.config` file opens.
4. Provide information in the fields provided. All the fields are described in *Appendix B, Traffic Manager Configuration Options*.
5. Click the **Add** button to add the mapping rule to the file and then click the **Apply** button to save your changes.
6. Click the **Close** button to close the configuration file editor.

▼ **To create mapping rules manually:**

1. In a text editor, open the `remap.config` file located in the Traffic Edge `config` directory.
2. Enter your map and reverse-map rules; refer to *remap.config, on page 446*.
3. Save and close the `remap.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

Enabling HTTP Reverse Proxy

You can enable HTTP reverse proxy either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ **To enable HTTP reverse proxy from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Protocols** button and then click the **HTTP** button.
3. Click the **Cacheability** tab.
4. Make sure that the **HTTP Caching** option is enabled.
5. On the **Configure** tab, click the **Content Routing** button and then click the **Reverse Proxy** button.
6. On the **General** tab, enable the **Reverse Proxy** option.
7. Click the **Apply** button.

▼ **To enable HTTP reverse proxy manually:**

1. In a text editor, open the `records.config` file located in the `config` directory.
2. Edit the following variable:

Variable	Description
proxy.config.reverse_proxy.enabled	Set this variable to 1 to enable HTTP reverse proxy mode.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

Setting Optional HTTP Reverse Proxy Options

Traffic Edge provides several configuration options for reverse proxy that let you:

- Configure Traffic Edge to retain the client host header information in a request during translation
- Configure Traffic Edge to serve requests only to the origin servers listed in the mapping rules; requests to origin servers not listed in the mapping rules are not served
- Specify an alternate URL, to which incoming requests from older clients that do not provide Host headers are directed

You can set the optional reverse proxy configuration options either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ **To set optional HTTP reverse proxy options from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Content Routing** button and then click the **Mapping and Redirection** button.
3. In the **Serve Mapped Hosts Only** area on the **General** tab, select **Required** if you want Traffic Edge to serve requests only to the origin servers listed in the mapping rules. This option provides added security for your Traffic Edge system.
4. Enable the **Retain Client Host Header** option if you want to retain the client host header in a request during translation.
5. In the **Redirect No-Host Header to URL** field, enter an alternate URL to which incoming requests from older clients that do not provide a host header are directed.
6. Click the **Apply** button.

▼ **To set optional HTTP reverse proxy options manually:**

1. In a text editor, open the `records.config` file located in the `config` directory.
2. Edit the following variables:

Variable	Description
proxy.config.url_remap.pristine_host_hdr	Set this variable to 1 to retain the client host header in the request. Set this variable to 0 (zero) if you want Traffic Edge to translate the client host header.
proxy.config.url_remap.remap_required	Set this variable to 1 if you want Traffic Edge to serve requests only to the origin servers listed in the mapping rules of the `remap.config` file. Set this variable to 0 (zero) if you want Traffic Edge to serve requests to all origin servers.
proxy.config.header.parse.no_host_url_redirect	Enter the URL to which to redirect requests with no host headers.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

Redirecting HTTP Requests

You can configure Traffic Edge to redirect HTTP requests without having to contact any origin servers; for example, if you redirect all requests for `http://www.ultraseek.com` to `http://www.server1.com/products/portal/search/`, all HTTP requests for `www.ultraseek.com` go directly to `www.server1.com/products/portal/search`.

You can configure Traffic Edge to perform permanent or temporary redirects. Permanent redirects notify the browser of the URL change (by returning an HTTP status code 301) so that the browser can update bookmarks. Temporary redirects notify the browser of the URL change for the current request only (by returning an HTTP status code 307).

You can set redirect rules either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ **To set redirect rules from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Content Routing** button and then click the **Mapping and Redirection** button.
3. In the **URL Remapping Rules** section, click the **Edit File** button.

   The configuration file editor for the `remap.config` file opens.

4 Enter the following information in the fields provided and then click the **Add** button.

- From the **Rule Type** drop-down list, select **redirect** or **temporary_redirect**.
- From the **Scheme** drop-down list, select **http** or **https**.
- In the **From Host** field, enter the hostname of the URL you want to redirect from.

Optional
- In the **From Port** field, enter the port number of the URL you want to redirect from.

Optional
- In the **From Path Prefix** field, enter the path prefix of the URL you want to redirect from.
- In the **To Host** field, enter the hostname of the URL you want to redirect to.

Optional
- In the **To Port** field, enter the port number of the URL you want to redirect to.

Optional
- In the **To Path Prefix** field, enter the path prefix of the URL you want to redirect to.

5 Click the **Apply** button to save your changes and then click the **Close** button to close the configuration file editor.

▼ To set redirect rules manually:

1 In a text editor, open the `remap.config` file located in the Traffic Edge `config` directory.

2 Enter a mapping rule for each redirect you want to set. Each mapping rule must be on a separate line and must consist of three space-delimited fields: `type`, `target`, and `replacement`. The following table describes the format for each field.

Field	Description
type	Enter either one of the following: `redirect`—redirects HTTP requests permanently without having to contact the origin server. `redirect_temporary`—redirects HTTP requests temporarily without having to contact the origin server.
target	Enter the origin or *from* URL. You can enter up to four components: *`scheme`*`://`*`host`*`:`*`port`*`/`*`path_prefix`*
replacement	Enter the destination or *to* URL. You can enter up to four components: *`scheme`*`://`*`host`*`:`*`port`*`/`*`path_prefix`*

The following example permanently redirects all HTTP requests for `www.server1.com` to `www.server2.com`.

```
redirect http://www.server1.com http://www.server2.com
```

3 Save and close the `remap.config` file.

4 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

5 Run the command `traffic_line -x` to apply the configuration changes.

FTP Reverse Proxy

In FTP reverse proxy mode, Traffic Edge receives FTP requests from FTP clients on behalf of an FTP server (typically, the FTP server hostname resolves to the Traffic Edge IP address).

Figure 22 illustrates how Traffic Edge serves an FTP request from an FTP client in reverse proxy mode.

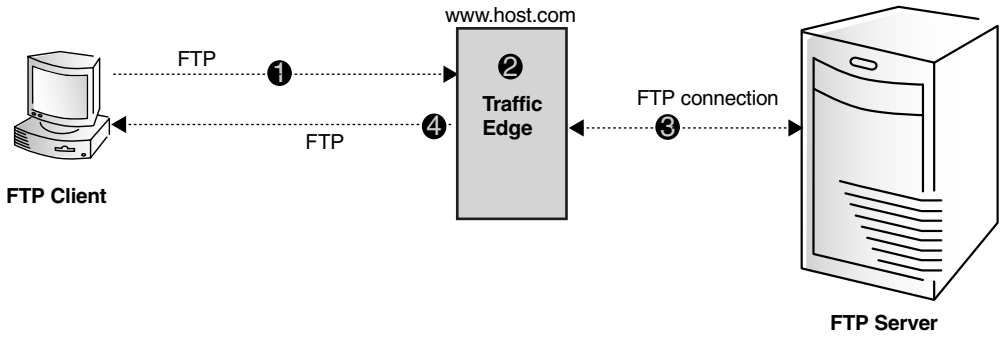


Figure 22 FTP reverse proxy

Figure 22 demonstrates the following steps:

1. An FTP client sends an FTP request to a host called `www.host.com`. Traffic Edge receives the request because it is acting as the FTP server (the advertised hostname of the FTP server resolves to Traffic Edge).
2. Traffic Edge locates a map rule in the `ftp_remap.config` file and remaps the request to the specified FTP server (`realhost.com`).
3. Traffic Edge opens an FTP connection to the origin server.
4. If the request is a cache hit and the content is fresh, Traffic Edge sends the requested document to the FTP client via FTP; if not, Traffic Edge obtains the requested document from the FTP server, sends the document to the FTP client via an FTP connection and saves a copy in its cache.

To use FTP reverse proxy, you must perform the following tasks:

- Set FTP mapping rules in the `ftp_remap.config` file; refer to *Setting FTP Mapping Rules, on page 139*.
- Enable the FTP reverse proxy option; refer to *Enabling FTP Reverse Proxy, on page 140*.

As an optional task, you can modify FTP options (for example, you can change the FTP connection mode and inactivity timeouts); refer to *Modifying FTP Options, on page 141*.

Setting FTP Mapping Rules

You must set FTP mapping rules so that Traffic Edge can direct any incoming FTP requests to the FTP server if the requested document is a cache miss or is stale. You can set FTP mapping rules either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ **To set FTP mapping rules from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Content Routing** button and then click the **Mapping and Redirection** button.
3. Click the **FTP** tab.
4. In the **Remapping Rules** section, click the **Edit File** button.

   The configuration file editor for the `ftp_remap.config` file opens.
5. Enter information in the fields provided and then click the **Add** button. All the fields are described in *Appendix B, Traffic Manager Configuration Options*.
6. Click the **Apply** button to save your changes and then click **Close** to close the configuration file editor.

▼ **To set FTP mapping rules manually:**

1. In a text editor, open the `ftp_remap.config` file located in the Traffic Edge `config` directory.
2. Enter one mapping rule per line, in the following format:

   `Traffic_Server``:`*`port ftp_server`*`:`*`port`*

   `Traffic_Server` is the IP address or hostname assigned to Traffic Edge and *`ftp_server`* is the IP address or hostname assigned to the FTP server to which you want to redirect the FTP requests.

Note Because FTP requests do not include host headers, Traffic Edge cannot distinguish between different FTP servers. Therefore, if you are working with multiple FTP servers, you must have multiple IP addresses assigned to Traffic Edge.

3. Save and close the `ftp_remap.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

Enabling FTP Reverse Proxy

To enable the FTP reverse proxy option, you can either use Traffic Manager or edit a configuration file manually. Both procedures are provided below.

▼ To enable FTP reverse proxy from Traffic Manager:

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. Click the **Configure** tab.

   The **General** tab of the **My Proxy**/**Basic** button displays.
3. In the **Features** table, click the **FTP On** button under the **Protocols** section to enable FTP processing and then click the **Apply** button.
4. On the **Configure** tab, click the **Protocols** button and then click the **FTP** button.
5. Click the **Cacheability** tab.
6. Enable the **FTP Caching** option, so that Traffic Edge caches the FTP documents it serves, and then click the **Apply** button.

   If the **FTP Caching** option is disabled, Traffic Edge always forwards FTP requests from FTP clients to the FTP server and does not serve them from its cache (Traffic Edge acts as a proxy server for FTP requests).
7. On the **Configure** tab, click the **Content Routing** button and then click the **Reverse Proxy** button.
8. Click the **FTP** tab.
9. Enable the **Reverse Proxy** option.
10. Click the **Apply** button.

▼ To enable FTP reverse proxy manually:

1. In a text editor, open the `records.config` file located in the `config` directory.
2. Edit the following variables:

Variable	Description
proxy.config.ftp.ftp_enabled	Set this variable to 1 to enable FTP on your Traffic Edge. This variable must be enabled for Traffic Edge to process FTP requests.
proxy.config.ftp.reverse_ftp_enabled	Set this variable to 1 to enable the FTP reverse proxy option. Note: If this variable is set to 0 but the proxy.config.ftp.ftp_enabled variable (described above) is set to 1, Traffic Edge will serve FTP requests in forward proxy mode.
proxy.config.ftp.cache_enabled	Set this variable to 1 to enable FTP document caching for requests sent from an FTP client. Traffic Edge will cache the FTP documents it serves. Set this variable to 0 (zero) to disable FTP document caching for requests sent from an FTP client. Traffic Edge always forwards FTP requests to the FTP server and does not serve the requests from its cache.

3 Save and close the `records.config` file.

4 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

5 Run the command `traffic_line -x` to apply the configuration changes.

Modifying FTP Options

After you have set FTP mapping rules and have enabled FTP reverse proxy, Traffic Edge can serve FTP requests in reverse proxy mode. Traffic Edge uses the default FTP options, such as the data connection mode and connection timeouts. You can modify the default FTP options to better suit your needs.

▼ **To modify FTP options from Traffic Manager:**

1 From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.

2 On the **Configure** tab, click the **Protocols** button and then click the **FTP** button.

3 In the **FTP Proxy Server Port** of the **General** tab, specify the port that Traffic Edge uses to accept FTP connections from FTP clients. The default port is 21.

4 In the **Listening Port Configuration** area, specify how FTP opens a listening port for a data transfer:

- Select **Default Settings** if you want the operating system to choose an available port. Traffic Edge sends 0 and retrieves the new port number if the listen succeeds.
- Select **Specify Range** if you want the listening port to be determined by the range of ports you specify in the **Listening Port (Max)** and **Listening Port (Min)** fields.

5 In the **Default Data Connection Method** area, specify the default method used to set up data connections with the FTP server:

- Select **Proxy Sends PASV** to send a PASV to the FTP server and let the FTP server open a listening port.
- Select **Proxy Sends PORT** to set up a listening port on the Traffic Edge side of the connection first.

6 Enable the **Shared Server Connections** option to share server control connections among multiple anonymous FTP clients.

7 Click the **Apply** button.

8 Click the **Cacheability** tab.

9 In the **Directory Caching** area:

- Enable the **Simple** option to cache directory listings without arguments: for example, `dir/ls`.
- Enable the **Full** option to cache directory listings with arguments: for example, `ls -al`, ls *.txt.

10 In the **Freshness** area, modify the freshness options to meet your needs:

- In the **Login Information** field, enter the number of seconds that 220/230 responses (login messages) can stay fresh in the cache.
- In the **Directory-Listing** field, enter the number of seconds that directory listings can stay fresh in the cache.
- In the **Files** field, enter the number of seconds that FTP files can stay fresh in the cache.

11 Click the **Apply** button.

12 Click the **Timeouts** tab.

13 Change the timeout values according to your needs.

14 Click the **Apply** button.

▼ **To modify FTP options manually:**

1 In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.

2 Go to the `FTP Engine` section of the file.

3 Edit the configuration variables in this section to suit your needs. The variables are described in *FTP Engine, on page 420*.

4 Save and close the `records.config` file.

5 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

6 Run the command `traffic_line -x` to apply the configuration changes.

Streaming Media Reverse Proxy

Traffic Edge receives streaming media requests from media players on behalf of a media server (typically, the media server hostname resolves to the Traffic Edge IP address).

This section provides the following information:

- An overview of reverse proxy for QuickTime requests; refer to *Reverse Proxy for QuickTime Requests, on page 143*.
- An overview of reverse proxy for Real media player requests; refer to *Reverse Proxy for Real Media Player Requests, on page 144*.
- An overview of reverse proxy for WMT requests; refer to *Reverse Proxy for WMT Requests, on page 145*.
- An overview of the steps needed to configure reverse proxy for streaming media; refer to *Configuring Streaming Media Reverse Proxy, on page 147*.

Reverse Proxy for QuickTime Requests

Figure 23 illustrates reverse proxy for QuickTime requests.

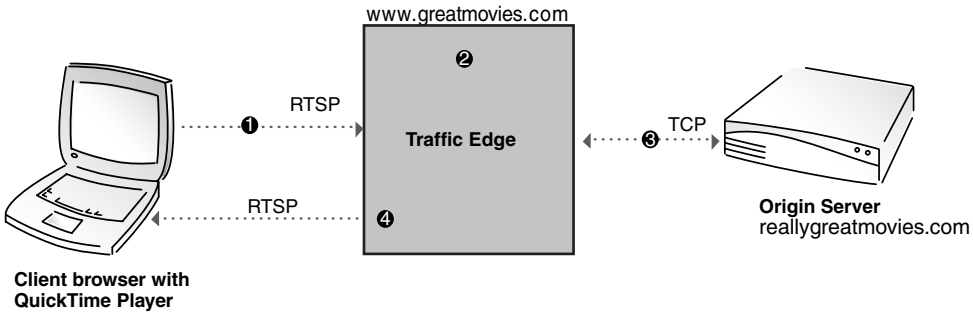


Figure 23 Reverse proxy for QuickTime requests

Figure 23 demonstrates the following steps:

1. A QuickTime Player sends an RTSP request addressed to a host called `www.greatmovies.com` on port `554`. Traffic Edge receives the request because it is acting as the media server.
2. Traffic Edge locates a map rule in the `remap.config` file and remaps the request to the specified origin server (`reallygreatmovies.com`).
3. Traffic Edge opens a TCP connection to the origin server and obtains a last-modified time for the requested stream.
4. Traffic Edge determines if the requested stream is in the cache and if it is fresh (the last-modified time obtained from the origin server must match the last-modified time in the cached stream). If so, Traffic Edge serves the requested stream; if not, Traffic Edge retrieves the stream from the QuickTime server, stores the stream in the cache and simultaneously serves it to the client.

Reverse Proxy for Real Media Player Requests

IMPORTANT Reverse proxy is not supported for Real Networks PNA requests.

Figure 24 illustrates reverse proxy for Real media player requests.

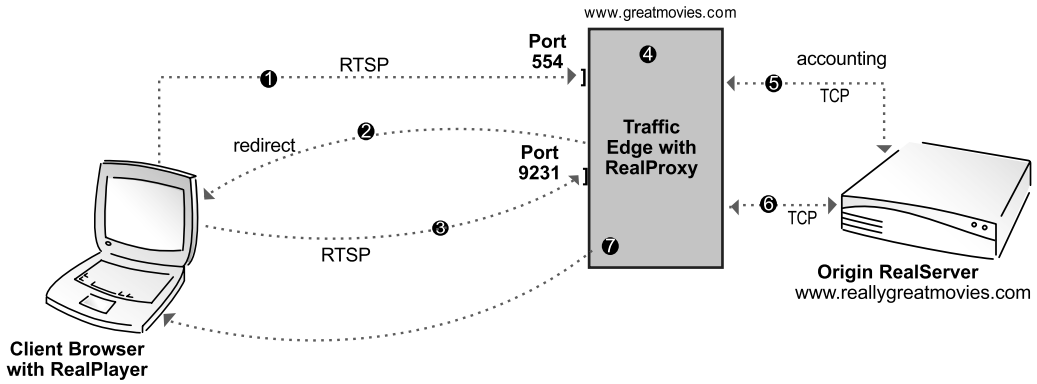


Figure 24 Reverse proxy for Real media player requests

Figure 24 demonstrates the following steps:

1. A Real media player sends an RTSP request addressed to `www.greatmovies.com` on port 554. Traffic Edge receives the request because it is acting as the origin RealServer.
2. Because the RTSP request comes from a Real media player, the Traffic Edge issues the client a redirect to port 9231.

Note For RealOne Player requests, Traffic Edge does not issue a redirect to port 9231; instead Traffic Edge establishes a tunnel to 9231 (the RealProxy port).

3. The Real media player sends the redirected RTSP request to the Traffic Edge RealProxy on port 9231.
4. Traffic Edge locates the map rule and remaps the request to the specified origin RealServer (`www.reallygreatmovies.com`).
5. RealProxy opens a TCP accounting connection on port 554 to the origin RealServer, dedicating the connection to the client that sent the request.
6. If the client is allowed access, RealProxy opens a TCP data connection to the origin RealServer; if not, RealProxy closes the accounting connection, returns an error to the client and the remaining steps do not apply.
7. RealProxy determines if the requested stream is in the cache and if it is fresh. If so, RealProxy serves the stream to the client; if not, RealProxy retrieves the stream from the origin server on port 7878, stores the stream in the cache and simultaneously serves it to the client.

Reverse Proxy for WMT Requests

WMT streaming requires both a web (HTTP) server for the metafiles and an Windows Media Server for the streaming media content. You must decide if you want to provide reverse proxy caching for one or both servers. Each requires a separate set of mapping rules.

Figure 25 illustrates reverse proxy for a Windows Media Server. In this example, the Traffic Edge `remap.config` file contains one map rule.

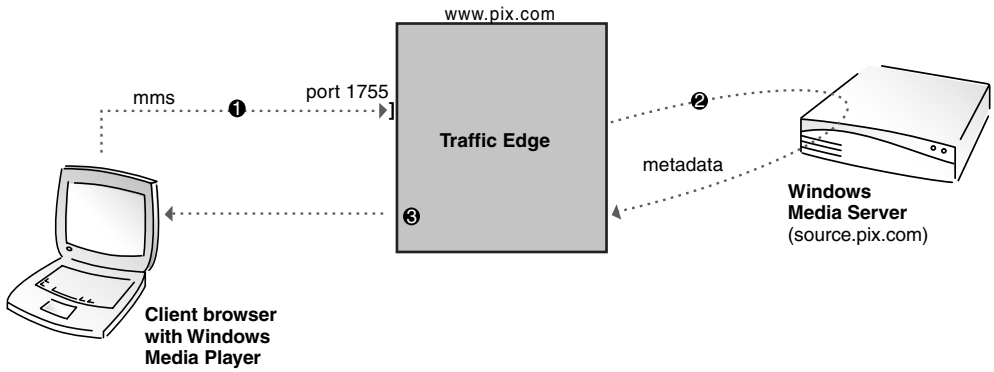


Figure 25 Reverse proxy for a Windows Media Server

Figure 25 demonstrates the following steps:

1. The Windows Media Player sends a request for a stream addressed to `www.pix.com` over MMS on port 1755. Traffic Edge receives the request because it is acting as the media server.
2. Using a map rule, Traffic Edge locates and opens a connection to the media server that contains the requested stream.
3. Traffic Edge attempts to find an exact match between the origin WMT server metadata and metadata in the Traffic Edge cache. If it does, the stream is a cache hit and Traffic Edge serves the requested stream to the client; if not, Traffic Edge retrieves the stream from the Windows Media Server, stores the stream in the cache and simultaneously serves the stream to the client.

Figure 26 illustrates reverse proxy for both a web server and a Windows Media Server. In this example, the Traffic Edge `remap.config` file contains two map rules and one reverse-map rule.

Note *Figure 26* illustrates an MMS request for a WMT stream. Traffic Edge also supports HTTP.

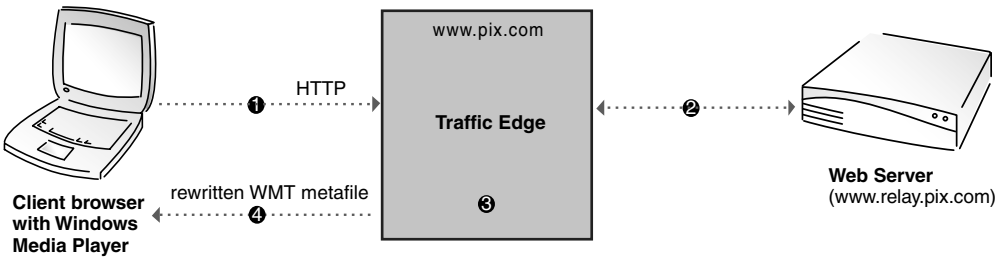


Figure 26 Reverse proxy for a web server and a media server

Figure 26 demonstrates the following steps:

1. The client sends a request for a WMT metafile to `www.pix.com` via HTTP on port 80. Traffic Edge receives the request because it is acting as the web server.
2. Traffic Edge locates and obtains the WMT metafile on the web server using a map rule (`map http://www.pix.com/ http://www.relay.pix.com/`).
3. Using the reverse-map rule, Traffic Edge rewrites the URL in the metafile to point to Traffic Edge (`reverse_map mms://www.source.pix.com/ mms://www.pix.com/`).
4. Traffic Edge forwards the rewritten WMT metafile to the client.

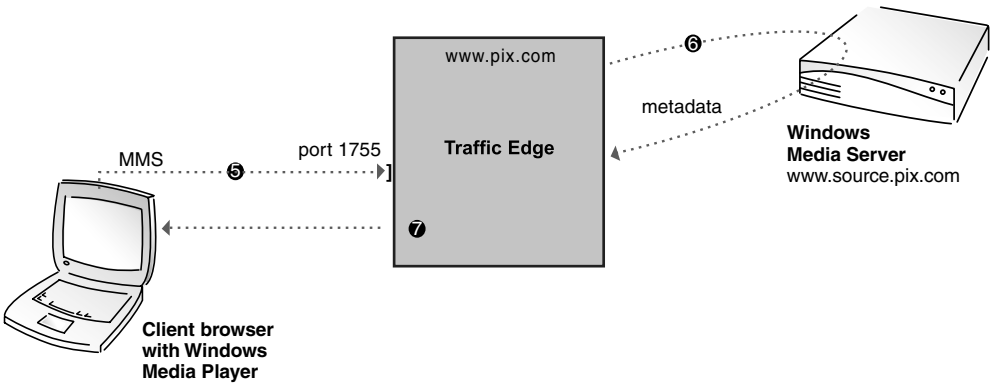


Figure 27 Reverse proxy for a web server and a WMT media server

Figure 27 demonstrates the following steps:

5. The Windows Media Player sends the request with the rewritten media content URL to Traffic Edge over MMS on port 1755.
6. Using the map rule (`map mms://www.pix.com/ mms://www.source.pix.com`), Traffic Edge opens a connection to the media server and obtains metadata about the requested stream.

7 Traffic Edge attempts to find an exact match between the origin WMT server metadata and metadata in the Traffic Edge cache. If it does, the stream is a cache hit and Traffic Edge serves the requested stream to the client; if not, Traffic Edge retrieves the stream from the Windows Media Server, stores the stream in the cache and simultaneously serves the stream to the client.

Configuring Streaming Media Reverse Proxy

To configure Traffic Edge to serve streaming media requests in reverse proxy mode, you must perform the following tasks:

- Set mapping rules in the `remap.config` file; refer to *Setting Mapping Rules for Streaming Media Requests*, below.
- Enable the reverse proxy option; refer to *Enabling Reverse Proxy, on page 150*.

Setting Mapping Rules for Streaming Media Requests

Traffic Edge uses two types of mapping rules for streaming media reverse proxy: *map* and *reverse-map* rules.

Both map and reverse-map rules consist of a *target* (origin) URL and a *replacement* (destination) URL. In a map rule, the target URL points to Traffic Edge and the replacement URL specifies where the original content is located. In a reverse-map rule, the target URL specifies where the original content is located and the replacement URL points to Traffic Edge. Traffic Edge stores mapping rules in the `remap.config` file located in the Traffic Edge `config` directory.

Note If two mappings match a request URL, Traffic Edge applies the first mapping listed in the `remap.config` file.

Traffic Edge does not support reverse-map rules for Real Networks.

You can set mapping rules either by using Traffic Manager or by editing a configuration file manually.

▼ **To set mapping rules from Traffic Manager:**

1 From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.

2 On the **Configure** tab, click the **Content Routing** button and then click the **Mapping and Redirection** button.

3 In the **URL Mapping Rules** section of the **General** tab, click the **Edit File** button.

The configuration file editor for the `remap.config` file opens.

4 Provide information in the fields provided:

- From the **Rule Type** drop-down box, select **map** to create a map rule or select **reverse_map** to create a reverse-map rule.
- From the **Scheme** drop-down list box, select **rtsp** for QuickTime or Real Networks, or select **mms** for WMT.
- In the **From Host** field for a map rule, enter the hostname of the Traffic Edge node. For a reverse-map rule, enter the hostname of the media server that contains the content.

Optional

- In the **From Port** field for a map rule, enter the port on which Traffic Edge accepts the particular requests; for example, the default port for RTSP (QuickTime and Real Networks) is 554 and the default port for MMS is 1755. For a reverse-map rule, enter the port on which the media server accepts the particular requests; for example, the default port for RTSP (QuickTime and Real Networks) is 554 and the default port for MMS is 1755.

Optional

- In the **From Path Prefix** field, enter the path prefix of the URL.
- From the **To Scheme** drop-down box, select **rtsp** for QuickTime or Real Networks, or select **mms** for WMT.
- In the **To Host** field for a map rule, enter the hostname of the media server that contains the content. For a reverse-map rule, enter the hostname of the Traffic Edge node.

Optional

- In the **To port** field for a map rule, enter the port on which the media server accepts the particular requests; for example, the default port for RTSP (QuickTime and Real Networks) is 554 and the default port for MMS is 1755. For a reverse-map rule, enter the port on which Traffic Edge accepts the particular requests; for example, the default port for RTSP (QuickTime and Real Networks) is 554 and the default port for MMS is 1755.

Optional

- In the **To Path Prefix** field, enter the path prefix of the URL.
- From the **MIXT Scheme** drop-down box, select **QT** if the map rule is for QuickTime requests or **RNI** if the map rule is for Real Networks requests.

5 Click the **Add** button to add the mapping rule to the file and then click the **Apply** button to save your changes.

6 Click the **Close** button to close the configuration file editor.

Note Traffic Edge supports HTTP for WMT requests in addition to MMS. For information about setting mapping rules for HTTP, refer to *Creating Mapping Rules for HTTP Requests, on page 132*.

▼ **To set mapping rules manually:**

1. In a text editor, open the `remap.config` file located in the Traffic Edge `config` directory.
2. Enter map rules in the following format:

   `map` *`scheme`*`://`*`traffic_edge`*`:`*`port/`* *`scheme`*`://`*`media_server`*`:`*`port/`* `QT|RNI`

   Enter reverse-map rules in the following format:

   `reverse_map` *`scheme`*`://`*`media_server`*`:`*`port/`* *`scheme`*`://`*`traffic_edge`*`:`*`port/`* `QT|RNI`

   `scheme` is the protocol used for the rule. Enter `rtsp` for QuickTime or Real Networks, or select `mms` for WMT.

   `traffic_edge:port` is the hostname of the Traffic Edge node and the port on which Traffic Edge listens for the particular requests; for example, the default port for RTSP (QuickTime and Real Networks) is 554 and the default port for MMS is 1755.

   `media_server:port` is the hostname of the media server that contains the content and *`port`* is the port on which the media server listens for the particular requests; for example, the default port for RTSP (QuickTime and Real Networks) is 554 and the default port for MMS is 1755.

   Specify `QT` for QuickTime or `RNI` for Real Networks.

   The following examples show map and reverse-map rules for QuickTime, Real Networks, and WMT:

   ```
   map rtsp://proxy1.domain.com:554/ rtsp://media1.domain.com:554/ QT
   reverse_map rtsp://media1.domain.com:554/ rtsp://proxy1.domain.com:554/ QT
   map rtsp://proxy.domain.com:554/ rtsp:// media.domain.com:554/ RNI
   map mms://www.pix.com/ mms://source.pix.com:1755/
   reverse_map mms://source.pix.com:1755/ mms://www.pix.com/
   ```

IMPORTANT The trailing slashes in all the rules are required.

Note Traffic Edge does not support reverse-map rules for Real Networks.

3. Save and close the `remap.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

5. Run the command `traffic_line -x` to apply the configuration changes.

Enabling Reverse Proxy

To enable the reverse proxy option for streaming media, you can either use Traffic Manager or edit a configuration file manually. Both procedures are provided below.

▼ **To enable reverse proxy from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Content Routing** button and then click the **Reverse Proxy** button.
3. Enable the **Reverse Proxy** option.
4. Click the **Apply** button.

▼ **To enable reverse proxy manually:**

1. In a text editor, open the `records.config` file located in the `config` directory.
2. Edit the following variable:

Variable	Description
proxy.config.reverse_proxy.enabled	Set this variable to 1 to enable reverse proxy mode.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

Note When you enable reverse proxy mode, Traffic Edge remaps incoming requests according to the map rules in the `remap.config` file. In addition, Traffic Edge serves all requests that do not match a map rule in forward proxy mode. If you want to run in reverse proxy *only* mode, where Traffic Edge does not serve requests that do not match a map rule, you must set the configuration variable `proxy.config.url_remap.remap_required` to 1 in the `records.config` file.

CHAPTER 8

Traffic Edge Clusters

Traffic Edge scales from a single node to multiple nodes that form a cluster allowing you to improve system performance and reliability.

This chapter discusses the following topics:

Understanding Traffic Edge Clusters

A Traffic Edge cluster consists of multiple Traffic Edge nodes. The nodes in a cluster share configuration information and can form a single logical cache.

Traffic Edge detects the addition and deletion of nodes in the cluster automatically and can detect when a node is unavailable. When the *virtual IP failover* option (described in *Using Virtual IP Failover, on page 155*) is enabled, the live nodes in a cluster can assume a failed node's responsibilities.

Traffic Edge uses a proprietary protocol for clustering, which is multicast for node location and heartbeat, but unicast for all data exchange within the cluster.

Traffic Edge has two clustering modes:

- Management-only mode; refer to *Management-Only Clustering* below.
- Full-clustering mode; refer to *Full Clustering, on page 152*.

IMPORTANT In a proxy hierarchy, the nodes in the cluster cannot be ICP siblings or be a mixture of HTTP parents and children; you must configure each node in a Traffic Edge cluster as a single node in the hierarchy because they share a common configuration.

Note Traffic Edge supports management-only clustering mode for streaming media. Full-clustering mode is not supported.

Management-Only Clustering

In management-only clustering mode, Traffic Edge cluster nodes share configuration information. You can administer all the nodes at the same time.

Traffic Edge uses a multicast management protocol to provide a single system image of your Traffic Edge cluster. Information about cluster membership, configuration, and exceptions is shared across all nodes, and the `traffic_manager` process automatically propagates configuration changes to all the nodes.

Full Clustering

In full-clustering mode, as well as sharing configuration information, a Traffic Edge cluster distributes its cache across its nodes into a single, virtual object store, rather than replicating the cache node by node. Traffic Edge can provide an enormous aggregate cache size and can maximize cache hit rate by storing objects only once across the entire cluster.

A fully clustered Traffic Edge maps objects to specific nodes in the cluster. When a node receives a request, it checks to see if the request is a hit somewhere in the cluster. If the request is a hit on a different node, the node handling the request obtains the object from the hit node and serves it to the client. Traffic Edge uses a proprietary communication protocol to obtain an object from sibling cluster nodes.

If a node fails or is shut down and removed, Traffic Edge removes references to the missing node on all nodes in the cluster. If virtual IP failover (described in *Using Virtual IP Failover, on page 155*) is enabled, requests destined for the missing node are handled by another node.

Full clustering requires a dedicated network interface for cluster communication.

Changing Clustering Mode

You can change the clustering mode either by using Traffic Manager or by editing a configuration file manually. Both procedures are described below.

Traffic Edge does not apply the clustering mode change to all the nodes in the cluster. You must change the clustering mode on each node individually.

▼ **To change clustering mode from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. Click the **Configure** tab.

   The **General** tab of the **My Proxy/Basic** button displays.
3. Click the **Clustering** tab.
4. In the **Cluster Type** area, select the clustering mode:
   - Select **Single Node** if this Traffic Edge node is not part of a cluster.
   - Select **Management Clustering** if you want your Traffic Edge cluster nodes to share configuration information only.
   - Select **Full Cache Clustering** if you want to both share configuration information between the cluster nodes and to distribute the Traffic Edge cache across all the cluster nodes into a single, virtual object store.

IMPORTANT To use either management-only clustering or full clustering, the Traffic Edge node must have second interface card.

5. Click the **Apply** button.

▼ **To change clustering mode manually:**

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Edit the following variable:

Variable	Description
proxy.local.cluster.type	Set this variable to: 1 for full-clustering mode. 2 for management-only mode. 3 for no clustering.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

Adding and Deleting Nodes in a Cluster

You can add a node or delete a node from a Traffic Edge cluster at any time. When you add a new node to the cluster, Traffic Edge detects it automatically. When you remove a node from the cluster, Traffic Edge removes all references to the missing node.

Adding Nodes to a Cluster

Traffic Edge can automatically detect new Traffic Edge nodes on your network and add them to the cluster, propagating the latest configuration information to the newcomer. This provides a convenient way to bootstrap new machines.

To connect an additional node to a Traffic Edge cluster, you need only install Traffic Edge software on the new node, making sure that the cluster name and port assignments are the same as those of the existing cluster. Traffic Edge automatically recognizes the new node.

IMPORTANT The nodes in a cluster must be homogeneous; each node must be the same hardware platform and must run the same version of the same operating system.

▼ **To add a node to a cluster:**

1. Install the appropriate hardware and connect it to your network. Consult your hardware documentation for hardware installation instructions.
2. Install the Traffic Edge software using the appropriate procedure for installing a cluster node; refer to the *Traffic Edge Installation Guide*. During installation, make sure that the following is true:
   - The cluster name that you assign to the new node is the same as the cluster name for the existing cluster.
   - The port assignments for the new node are the same as the port assignments used by the other nodes in the cluster.

3 Start Traffic Edge; refer to *Starting Traffic Edge, on page 25*.

If you have an existing Traffic Edge installation and you want to add that Traffic Edge to the cluster, you do *not* have to reinstall the Traffic Edge software on the node. Instead, you can edit certain configuration variables on the existing Traffic Edge node. Use the following procedure.

▼ To add an existing Traffic Edge installation to a cluster:

1 In a text editor, open the `records.config` file located in the `config` directory on the node you want to add to the cluster.

2 Edit the following variables:

Variable	Description
proxy.config.cluster.type	Set this variable to specify the clustering mode used by the other nodes in the cluster: 1 for full-clustering mode 2 for management-only mode Clustering modes are described in *Understanding Traffic Edge Clusters, on page 151*.
proxy.config.proxy_name	Set this variable to the name of Traffic Edge cluster. All nodes in a cluster must use the same name.
proxy.config.cluster.mc_group_addr	Set this variable to specify the multicast address for cluster communications. All nodes in a cluster must use the same multicast address.
proxy.config.cluster.rsport	Set this variable to specify the reliable service port. The reliable service port is used to send data between the nodes in the cluster. All nodes in a cluster must use the same reliable service port. The default value is 8088.
proxy.config.cluster.mcport	Set this variable to specify the multicast port. The multicast port is used for node identification. All nodes in a cluster must use the same multicast port. The default port number is 8089.
proxy.config.cluster.ethernet_interface	Set this variable to specify the network interface for cluster traffic. All nodes in a cluster must use the same network interface.

3 Save and close the `records.config` file.

4 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

5 Run the command `traffic_line -L` to restart the `traffic_manager` process on the local node.

Deleting Nodes from a Cluster

To delete a node from the Traffic Edge cluster, you need to edit a configuration variable on the node you want to delete. Use the following procedure.

▼ **To delete a node from a cluster:**

1. Stop Traffic Edge on the node you want to delete.
2. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
3. Edit the following variable:

Variable	Description
proxy.config.cluster.type	Set this variable to 3 to turn off clustering.

4. Save and close the `records.config` file.
5. Restart Traffic Edge.

Using Virtual IP Failover

The Traffic Edge virtual IP failover option enables Traffic Edge to maintain a pool of virtual IP addresses that it assigns to the nodes in the cluster as necessary. These virtual IP addresses are virtual only in the sense that they are not tied to a specific machine; Traffic Edge can assign them to any of its nodes. To the outside world, these virtual IP addresses are *the* addresses of the Traffic Edge cluster.

Virtual IP failover assures that if a node in the cluster fails, other nodes can assume the failed node's responsibilities. Traffic Edge handles virtual IP failover in the following ways:

- The `traffic_manager` process maintains cluster communication. Nodes automatically exchange statistics and configuration information through multicast communication. If multicast heartbeats are not received from one of the cluster nodes, the other nodes recognize it as unavailable.
- The `traffic_manager` process reassigns the IP addresses of the failed node to the remaining operational nodes within approximately 30 seconds, so that service can continue without interruption.
- The IP addresses are assigned to new network interfaces and the new assignment is broadcast to the local network. The IP reassignment is done through a process called *ARP rebinding*.

What Are Virtual IP Addresses?

Virtual IP addresses are really just IP addresses. They are called virtual addresses because they are not tethered to particular machines and can rotate among nodes in a Traffic Edge cluster.

It is common for a single machine to represent multiple IP addresses on the same subnet. This machine would have a primary or real IP address bound to its interface card and also serve many more virtual addresses.

You can set up your user base to use a DNS round-robin pointing at virtual IP addresses, as opposed to using the real IP addresses of the Traffic Edge machines.

Because virtual IP addresses are not bound to machines, a Traffic Edge cluster can *steal* addresses from inactive Traffic Edge nodes and distribute those addresses among the remaining live nodes. Using a proprietary Inktomi management protocol, Traffic Edge nodes communicate their status with their peers. If a node fails, its peers notice the failure and quickly negotiate which of the remaining nodes will mask the fault by taking over the failed node's virtual interface.

Setting Virtual IP Address Options

Traffic Edge provides several configuration options for virtual IP addressing:

- You can enable and disable virtual IP addressing; refer to *Enabling and Disabling Virtual IP Addressing*, below.
- You can add, modify, and delete virtual IP addresses; refer to *Adding and Editing Virtual IP Addresses, on page 157*.

Enabling and Disabling Virtual IP Addressing

You can turn virtual IP addressing on or off either by using Traffic Manager or by editing a configuration file manually. Both procedures are described below.

▼ **To enable or disable virtual IP addressing from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. Click the **Configure** tab.

   The **General** tab of the **My Proxy**/**Basic** button displays.
3. In the **Features** table, click the **Virtual IP** button under the **Networking** section to enable or disable Virtual IP addressing.
4. Click the **Apply** button.
5. Click the **Restart** button to restart Traffic Edge on all the nodes in the cluster.

▼ **To enable or disable virtual IP addressing manually:**

1. In a text editor, open the `records.config` file located in the `config` directory.
2. Edit the following variable:

Variable	Description
proxy.config.vmap.enabled	Set this variable to 1 to enable virtual IP addressing. Set this variable to 0 (zero) to disable virtual IP addressing.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -M` to restart the `traffic_manager` process on all the nodes in the cluster.

Adding and Editing Virtual IP Addresses

You can add new or edit existing virtual IP addresses from Traffic Manager.

CAUTION Incorrect IP addressing can effectively disable your system. Make sure you understand how virtual IP addresses work before changing them.

Virtual IP addresses must be prereserved like all IP addresses before they can be assigned to Traffic Edge.

▼ **To add or edit virtual IP addresses:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Networking** button and then click the **Virtual IP** button.

   The **Virtual IP Addresses** area displays the virtual IP addresses managed by Traffic Edge.

   The **Virtual IP** button displays only if you have enabled the **Virtual IP** option in the **Features** table under **My Proxy**/**Basic**/**General**.
3. Click the **Edit File** button to add new or edit the existing virtual IP addresses.

   The configuration file editor opens.
4. To edit a virtual IP address, select it from the table at the top of the page, edit the fields provided and then click the **Set** button.

   To delete the selected IP address, click the **Clear Fields** button.

   To add a virtual IP address, specify the virtual IP address, the Ethernet interface, and the sub-interface in the fields provided and then click the **Add** button.

   The **Sub-interface** field specifies the sub-interface ID. This is a number between 1 and 255 that the interface uses for the address. The sub-interface ID must be different for each specified virtual IP address; otherwise, clustering fails.
5. Click the **Apply** button to save your changes and then click the **Close** button to close the configuration file editor.
6. Restart Traffic Edge by clicking the **Restart** button on the **Configure**/**My Proxy**/**Basic/General** tab.

CHAPTER 9 Hierarchical Caching

Traffic Edge can participate in cache hierarchies, where requests not fulfilled in one cache can be routed to other regional caches, taking advantage of the contents and proximity of nearby caches.

This chapter discusses the following topics:

Understanding Cache Hierarchies

A cache hierarchy consists of levels of caches that communicate with each other. Traffic Edge supports several types of cache hierarchies. All cache hierarchies recognize the concept of *parent* and *child*. A parent cache is a cache higher up in the hierarchy, to which Traffic Edge can forward requests. A child cache is a cache for which Traffic Edge is a parent.

Traffic Edge supports the following hierarchical caching options:

- Parent caching; refer to *Parent Caching*, below.
- ICP (Internet Cache Protocol) peering; refer to *ICP Peering, on page 163*.

Parent Caching

If a Traffic Edge node cannot find a requested object in its cache, it can search a parent cache, which itself can search other caches, before resorting to retrieving the object from the origin server.

You can configure a Traffic Edge node to use one or more parent caches. You use more than one parent cache so that if one parent is unavailable, another parent can service requests. This is called *parent failover* and is described in *Parent Failover, on page 160*.

Note If you do not want all requests to go to the parent cache, you can configure Traffic Edge to route certain requests directly to the origin server; for example, requests that contain specific URLs, by setting parent proxy rules in the `parent.config` configuration file (described in *parent.config, on page 394*).

Traffic Edge supports parent caching for HTTP requests, FTP-over-HTTP requests, and QuickTime, Real Networks and WMT live and on-demand streaming media requests.

Figure 28 illustrates a simple cache hierarchy, in which a Traffic Edge node is configured to use a parent cache.

In this figure, a client sends a request to a Traffic Edge node (which is a child in the cache hierarchy because it is configured to forward missed requests to a parent cache). The request is a cache miss, so the Traffic Edge forwards the request to the parent cache. On the parent, the request is a cache hit, so the parent sends a copy of the content to the Traffic Edge, where it is cached and then served to the client. (Future requests for this content can now be served directly from the Traffic Edge cache.)

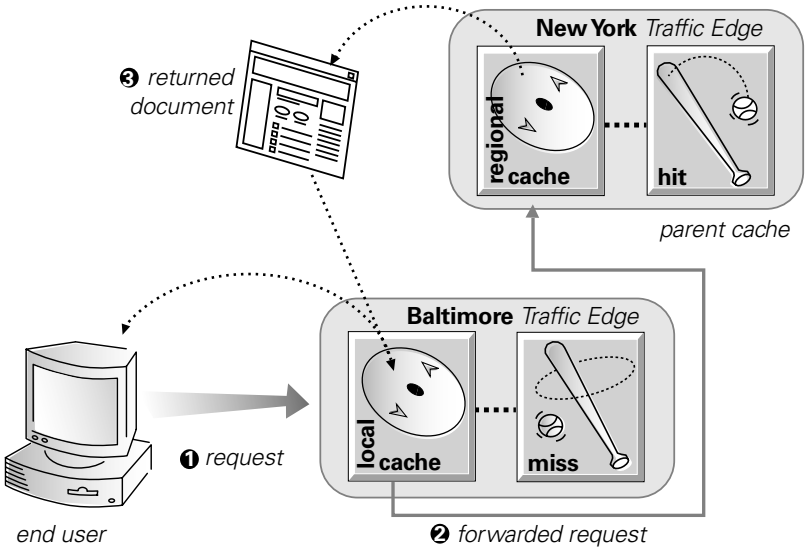


Figure 28 Parent caching

Note If the request is a cache miss on the parent, the parent retrieves the content from the origin server (or from another cache, depending on the parent's configuration). The parent caches the content and then sends a copy to the Traffic Edge (its child), where it is cached and served to the client.

Parent Failover

Traffic Edge supports the use of several parent caches so that if one parent cache is not available, another parent cache can service client requests.

When you configure your Traffic Edge to use more than one parent cache, Traffic Edge detects when a parent is not available and sends missed requests to another parent cache. If you specify more than two parent caches, the order in which the parent caches are queried depends upon the parent proxy rules configured in the parent configuration file described in *parent.config, on page 394*. By default, the parent caches are queried in the order in which they are listed in the configuration file.

Configuring Traffic Edge to Use a Parent Cache

To configure Traffic Edge to use one or more parent caches, you must complete the following steps:

- Enable the parent caching option.
- Identify the parent cache you want to use to service missed requests. To use *parent failover*, you must identify more than one parent cache so that when a parent cache is unavailable, requests are sent to another parent cache.

You can identify parent caches and enable the parent caching option either by using Traffic Manager or by editing configuration files manually. Both procedures are provided below:

Note You need to configure the child cache only. No additional configuration is needed on the Traffic Edge parent cache.

▼ **To configure Traffic Edge to use a parent cache from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Content Routing** button and then click the **Hierarchies** button.
3. On the **Parenting** tab, enable the **Parent Proxy** option and click the **Apply** button.
4. In the **Parent Proxy Cache Rules** area, click the **Edit File** button.

   The configuration file editor for the `parent.config` file opens.
5. From the **Primary Destination Type** drop-down box, select the type of rule you want to create:
   - Select **dest_domain** to create a rule for requests that contain a particular destination domain
   - Select **dest_host** to create a rule for requests that contain a particular destination host
   - Select **dest_ip** to create a rule for requests that contain a particular destination IP address
   - Select **url_regex** to create a rule for requests that contain a particular regular expression in the URL
6. In the **Primary Destination Value** field, enter the value for the primary destination you selected from the **Primary Destination Type** drop-down box; for example, if you selected **dest_ip**, enter the destination IP address for which you want to create the rule.
7. In the **Parent Proxies** field, enter a list of the parent caches that will serve requests. Separate each parent cache with a semicolon.
8. From the **Round Robin** drop-down list box, select **true** if you want Traffic Edge to go through the parent cache list in a round-robin based on client IP address. Select **strict** if you want Traffic Edge nodes to serve requests strictly in turn: for example, machine `proxy1` serves the first request, `proxy2` serves the second request, and so on. Select **false** if you do not want round-robin selection to occur.

9 From the **Go direct** drop-down list box, select **true** if you want Traffic Edge to go directly to the origin server and bypass the parent caches. Select **false** if you do not want Traffic Edge to bypass the parent caches.

10 From the **Secondary Specifiers** area, enter any optional information. All fields are described in *Hierarchies, on page 319*.

11 Click the **Add** button to add the parenting rule to the file and then click the **Apply** button to save your changes.

12 Click the **Close** button to close the configuration file editor.

▼ To configure Traffic Edge to use a parent cache manually:

1 In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.

2 Edit the following variable:

Variable	Description
proxy.config.http.parent_proxy_routing_enable	Set this variable to 1 to enable the parent caching option.

3 Save and close the `records.config` file.

4 In a text editor, open the `parent.config` file located in the Traffic Edge `config` directory.

5 Set parent proxy rules to specify the parent cache to which you want missed requests to be forwarded; refer to *parent.config, on page 394*.

The following example configures Traffic Edge to route all requests containing the regular expression `politics` and the path `/viewpoint` directly to the origin server (bypassing any parent hierarchies):

```
url_regex=politics prefix=/viewpoint go_direct=true
```

The following example configures Traffic Edge to direct all missed requests with URLs beginning with `mms://host1` to the parent cache `parent1`. If `parent1` cannot serve the requests, they are forwarded to `parent2`. Because `round-robin=true`, Traffic Edge goes through the parent cache list in a round-robin based on client IP address.

```
dest_host=host1 scheme=mms parent="parent1;parent2" round-robin=strict
```

6 Save and close the `parent.config` file.

7 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

8 Run the command `traffic_line -x` to apply the configuration changes.

ICP Peering

The Internet Cache Protocol (ICP) is a protocol used by proxy caches to exchange information about their content. ICP query messages ask other caches if they are storing a particular URL. ICP response messages reply with a hit or miss answer.

A cache exchanges ICP messages only with specific ICP peers, which are neighboring caches that can receive ICP messages. An ICP peer can be a sibling cache, which is at the same level in the hierarchy, or a parent cache, which is one level up in the hierarchy.

If Traffic Edge has ICP caching enabled, it sends out ICP queries to its ICP peers in the event of a cache miss on an HTTP request. If there are no hits and parents exist, a parent is selected using a round-robin policy. If no ICP parents exist, Traffic Edge forwards the request to its HTTP parents. If there are no HTTP parent caches established, Traffic Edge forwards the request to the origin server.

If Traffic Edge receives a hit message from an ICP peer, Traffic Edge sends the HTTP request to that peer. However, it might turn out to be a cache miss, because the original HTTP request contains header information that is not communicated by the ICP query: for example, the hit might not be the requested alternate. If an ICP hit turns out to be a miss, Traffic Edge forwards the request to either its HTTP parent caches or to the origin server.

To configure a Traffic Edge node to be part of an ICP cache hierarchy, you must perform the following tasks:

- Determine if the Traffic Edge can receive ICP messages only or both send and receive ICP messages.
- Determine if Traffic Edge can send messages directly to each ICP peer or send a single message on a specified multicast channel.
- Specify the port used for ICP messages.
- Set the ICP query timeout.
- Identify the ICP peers (siblings and parents) with which Traffic Edge can communicate.

You can configure a Traffic Edge node to be part of an ICP cache hierarchy either by using Traffic Manager or by editing configuration files manually. Both procedures are provided below.

▼ **To configure Traffic Edge to use an ICP cache hierarchy from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Content Routing** button and then click the **Hierarchies** button.
3. Click the **ICP Peering** tab.
4. In the **ICP mode** area, select:
   - **Only Receive Queries** to configure Traffic Edge to receive ICP queries from other ICP peers only. In this mode, Traffic Edge cannot send queries to other ICP peers.
   - **Send/Receive Queries** to configure Traffic Edge to both send and receive ICP queries.

5 In the **ICP Port** field, enter the port that you want to use for ICP messages. The default is 3130.

6 Enable the **ICP Multicast** option to send ICP messages through multicast if your Traffic Edge has a multicast channel connection to its ICP peers.

7 In the **ICP Query Timeout** field, enter the timeout for ICP queries. The default is 2 seconds.

8 In the **ICP Peers** section, click the **Edit File** button.

The configuration file editor for the `icp.config` file opens.

9 Enter information in the fields provided and then click the **Add** button. All the fields are described in *Appendix B, Traffic Manager Configuration Options*.

10 Click the **Apply** button to save the information and then click the **Close** button to exit the configuration file editor.

11 On the **ICP Peering** tab, click the **Apply** button to save your configuration.

▼ To configure Traffic Edge to use an ICP cache hierarchy manually:

1 In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.

2 Edit the following variables:

Variable	Description
proxy.config.icp.enabled	Set this variable to: 0 to disable ICP. 1 to allow Traffic Edge to receive ICP queries only. 2 to allow Traffic Edge to send and receive ICP queries.
proxy.config.icp.icp_port	Set this variable to specify the UDP port that you want to use for ICP messages. The default is 3130.
proxy.config.icp.multicast_enabled	Set this variable to: 0 to disable ICP multicast. 1 to enable ICP multicast.
proxy.config.icp.query_timeout	Set this variable to specify the timeout used for ICP queries. The default is 2 seconds.

3 Save and close the `records.config` file.

4 In a text editor, open the `icp.config` file located in the Traffic Edge `config` directory.

5 For each ICP peer you want to identify, enter a separate rule in the `icp.config` file; refer to *icp.config, on page 382*.

6 Save and close the `icp.config` file.

7 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

8 Run the command `traffic_line -x` to apply the configuration changes.

CHAPTER 10 Configuring the Cache

The Traffic Edge cache consists of a high-speed object database called the *object store* that indexes objects according to URLs and associated headers.

This chapter discusses the following topics:

The Traffic Edge Cache

The Traffic Edge cache consists of a high-speed object database called the *object store*. The object store indexes objects according to URLs and associated headers enabling Traffic Edge to store, retrieve, and serve not only web pages, but also parts of web pages, providing optimum bandwidth savings. Using sophisticated object management, the object store can cache alternate versions of the same object, varying on spoken language or encoding type, and can efficiently store very small and very large documents, minimizing wasted space. When the cache is full, Traffic Edge removes stale data, ensuring that the most requested objects are kept on-hand and fresh.

Traffic Edge is designed to tolerate total disk failures on any of the cache disks. If the disk fails completely, Traffic Edge marks the entire disk as corrupt and continues using the remaining disks. An alarm is sent to Traffic Manager, indicating which disk failed. If all of the cache disks fail, Traffic Edge goes into proxy-only mode.

You can perform the following cache configuration tasks:

- Change the total amount of disk space allocated to the cache; refer to *Changing Cache Capacity, on page 166*.
- Partition the cache by reserving cache disk space for specific protocols and origin servers/domains; refer to *Partitioning the Cache, on page 167*.
- Delete all data in the cache; refer to *Clearing the Cache, on page 170*.

IMPORTANT Traffic Edge does not use the cache to store streams for Real media player requests but uses the filesystem instead; refer to the *Traffic Edge Installation Guide*.

The RAM Cache

Traffic Edge maintains a small RAM cache of extremely popular objects. This RAM cache serves the most popular objects as fast as possible and reduces load on disks, especially during temporary traffic peaks. You can configure the RAM cache size to suit your needs; refer to *Changing the Size of the RAM Cache, on page 171*.

Changing Cache Capacity

You can increase or reduce the total amount of disk space allocated to the cache *without* clearing the content.

Tip You can check the size of the cache in bytes from Traffic Line by specifying the command `traffic_line -r proxy.process.cache.bytes_total`.

Increasing Cache Capacity

You can increase the total amount of disk space allocated to the cache on existing disks or add new disks to a Traffic Edge node.

▼ **To increase cache capacity:**

1. Stop Traffic Edge.
2. Add hardware, if necessary.
3. Edit the Traffic Edge `storage.config` file to increase the amount of disk space allocated to the cache on existing disks or to describe the new hardware you are adding; refer to *storage.config, on page 455*.

Linux only

4. If you add a new disk, you must edit the `/etc/rc.d/init.d/traffic_server` file to add a raw disk binding. The raw disk binding must have Inktomi user permissions. Instructions on adding a raw disk binding are located in the Traffic Edge `storage.config` file.
5. Restart Traffic Edge.

Reducing Cache Capacity

You can reduce the total amount of disk space allocated to the cache on an existing disk or remove disks from a Traffic Edge node.

▼ **To reduce cache capacity:**

1. Stop Traffic Edge.
2. Remove hardware, if necessary.
3. Edit the Traffic Edge `storage.config` file to reduce the amount of disk space allocated to the cache on existing disks or to delete the reference to the hardware you are removing; refer to *storage.config, on page 455*.

Linux only

4. If you remove a disk, you must edit the `/etc/rc.d/init.d/traffic_server` file to remove the raw disk binding for the disk.

5 Restart Traffic Edge.

IMPORTANT In the `storage.config` file, a formatted or raw disk must be at least 128 MB.

Partitioning the Cache

You can manage your cache space more efficiently and restrict disk usage by creating cache partitions of different sizes for specific protocols. You can further configure these partitions to store data from specific origin servers and/or domains.

IMPORTANT The partition configuration must be the same on all nodes in a cluster.

Creating Cache Partitions for Specific Protocols

You can create separate partitions for your cache that vary in size to store content according to protocol. This configuration ensures that a certain amount of disk space is always available for a particular protocol.

Traffic Edge supports two different partition types: *http* for HTTP and FTP objects and *mixt* for QuickTime and WMT streams. Traffic Edge provides these two partition types to support the different size and service requirements for streaming media objects. To take advantage of this cache partitioning, you must run Traffic Edge Media Edition.

Note Traffic Edge does not use the cache to store streams for Real media player requests but uses the filesystem instead; refer to the *Traffic Edge Installation Guide*.

You can partition the cache according to protocol either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ To partition the cache according to protocol from Traffic Manager:

1 From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.

2 On the **Configure** tab, click the **Subsystems** button and then click the **Cache** button.

3 Click the **Partition** tab.

4 In the **Cache Partition** area, click the **Edit File** button.

5 The configuration file editor for the `partition.config` file opens.

6 Enter information in the fields provided and then click the **Add** button. All the fields are described on *Cache, on page 335*.

7 Click the **Apply** button to save the information and then click the **Close** button to exit the configuration file editor.

8 Restart Traffic Edge by clicking the **Restart** button on the **Configure**/**My Proxy**/**Basic/General** tab.

▼ To partition the cache according to protocol manually:

1 In a text editor, open the `partition.config` file located in the Traffic Edge `config` directory.

2 Enter a line in the file for each partition you want to create; refer to *partition.config, on page 396*.

3 Save and close the `partition.config` file.

4 Restart Traffic Edge.

Making Changes to Partition Sizes and Protocols

After you have configured your cache partitions based on protocol, you can make changes to the configuration at any time. Before making changes, note the following:

- You must stop Traffic Edge before you change the cache partition size and protocol assignment.
- When you increase the size of a partition, the contents of the partition are *not* deleted However, when you reduce the size of a partition, the contents of the partition *are* deleted.
- When you change the partition number, the partition is deleted and then re-created, even if the size and protocol type remain the same.
- When you add new disks to your Traffic Edge node, the partition sizes specified in percentages increase proportionately.
- A lot of changes to the partition sizes might result in disk fragmentation, which affects performance and hit rate. Inktomi recommends that you clear the cache before making many changes to cache partition sizes; refer to *Clearing the Cache, on page 170*.

Partitioning the Cache According to Origin Server or Domain

After you have partitioned the cache according to size and protocol, you can assign the partitions you created to specific origin servers and/or domains.

You can assign a partition to a single origin server or multiple origin servers. However, if a partition is assigned to multiple origin servers, there is no guarantee on the space available in the partition for each origin server. Content is stored in the partition according to popularity.

In addition to assigning partitions to specific origin servers and domains, you must assign a generic partition to store content from all origin servers and domains that are not listed. This generic partition is also used if the partitions for a particular origin server or domain become corrupt.

IMPORTANT If you do not assign a generic partition, Traffic Edge will run in proxy-only mode.

Note You do *not* need to stop Traffic Edge before you assign partitions to particular hosts or domains. However, this type of configuration can cause a spike in memory usage and is time consuming. Inktomi recommends that you configure partition assignment during periods of low traffic.

You can partition the cache according to hostname and domain either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ To partition the cache according to hostname and domain from Traffic Manager:

1 From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.

2 Configure the cache partitions according to size and protocol, as described in *Creating Cache Partitions for Specific Protocols, on page 167*.

3. You should create a separate partition based on protocol for each host and domain, and an additional generic partition to use for content that does not belong to these origin servers or domains. For example, if you want to separate content from two different origin servers and you want the content to be separated by protocol so that HTTP content and streaming media content is stored separately, you must have five separate partitions: one HTTP-based partition for each origin server, one streaming media-based partition for each origin server, and a generic partition for all other origin servers not listed (the partitions do not have to be the same size).
4. On the **Configure** tab, click the **Subsystems** button and then click the **Cache** button.
5. Click the **Hosting** tab.
6. In the **Cache Hosting** area, click the **Edit File** button.
7. The configuration file editor for the `hosting.config` file opens.
8. Enter information in the fields provided and then click the **Add** button. All the fields are described in *Appendix B, Traffic Manager Configuration Options*.
9. Click the **Apply** button to save the information and then click the **Close** button to exit the configuration file editor.

▼ **To partition the cache according to hostname and domain manually:**

1. Configure the cache partitions according to size and protocol, as described in *Creating Cache Partitions for Specific Protocols, on page 167*.
2. You should create a separate partition based on protocol for each host and domain, and an additional generic partition to use for content that does not belong to these origin servers or domains. For example, if you want to separate content from two different origin servers and you want the content to be separated by protocol so that HTTP content and streaming media content is stored separately, you must have five separate partitions: one HTTP-based partition for each origin server, one streaming media-based partition for each origin server, and a generic partition for all other origin servers not listed (the partitions do not have to be the same size).
3. In a text editor, open the `hosting.config` file located in the Traffic Edge `config` directory.
4. Enter a line in the file to allocate the partition(s) used for each origin server and/or domain; refer to *hosting.config, on page 381*.
5. Assign a generic partition to use for content that does not belong to any of the origin servers or domains listed in the file. If all partitions for a particular origin server become corrupt, Traffic Edge will also use the generic partition to store content for that origin server; refer to *hosting.config, on page 381*.
6. Save and close the `hosting.config` file.
7. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
8. Run the command `traffic_line -x` to apply the configuration changes.

Configuring the Cache Object Size Limit

By default, Traffic Edge allows objects of any size in the cache. You can change the default behavior and specify a size limit for objects in the cache.

To specify a size limit for objects in the cache, you can either use Traffic Manager or edit a configuration file manually. Both procedures are provided below.

▼ **To specify a size limit for objects in the cache from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Subsystems** button and then click the **Cache** button.
3. In the **Maximum Object Size** field on the **General** tab, enter the maximum size allowed for objects in the cache in bytes. Enter 0 (zero) if you do not want to have a size limit.
4. Click the **Apply** button.

▼ **To specify a size limit for objects in the cache manually:**

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Edit the following variable:

Variable	Description
proxy.config.cache.max_doc_size	Set this variable to specify the maximum size allowed for objects in the cache in bytes. Enter 0 (zero) if you do not want to have a size limit.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

Clearing the Cache

When you clear the cache, you remove all data from the entire cache, which includes the data in the host database. You should clear the cache before performing certain cache configuration tasks, such as partitioning.

Note You cannot clear the cache when Traffic Edge is running.

▼ **To clear the cache:**

1. Stop Traffic Edge; refer to *Stopping Traffic Edge, on page 30*.
2. Enter the following command to clear the cache:

   ```
   traffic_server -Cclear
   ```

CAUTION

3. The `clear` command deletes all data in the object store and the host database. Traffic Edge does *not* prompt you to confirm the deletion.
4. Restart Traffic Edge; refer to *Starting Traffic Edge, on page 25*.

Changing the Size of the RAM Cache

The Traffic Edge provides a dedicated RAM cache for fast retrieval of popular small objects. The default RAM cache size is automatically calculated based on the number and size of the cache partitions you have configured. You can increase the RAM cache size for better cache hit performance.

CAUTION If you increase the size of the RAM cache and observe a decrease in Traffic Edge performance (such as increased latencies), the operating system might require more memory for network resources. Return the RAM cache size to its previous value.

Note If you have partitioned your cache according to protocol and/or hosts, the size of the RAM cache for each partition is proportional to the size of that partition.

▼ **To change RAM cache size from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Subsystems** button and then click the **Cache** button.
3. In the **Ram Cache Size** field on the **General** tab, enter the amount of space (in bytes) that you want to allocate to the RAM cache.

   The default value of -1 means that the RAM cache is automatically sized at approximately one MB per gigabyte of disk.
4. Click the **Apply** button.
5. Restart Traffic Edge by clicking the **Restart** button on the **Configure**/**My Proxy**/**Basic/General** tab.

▼ **To change RAM cache size manually:**

1. Stop Traffic Edge.
2. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
3. Edit the following variable:

Variable	Description
proxy.config.cache.ram_cache.size	Set this variable to specify the size of the RAM cache. The default value of -1 means that the RAM cache is automatically sized at approximately one MB per gigabyte of disk.

4. Save and close the `records.config` file.
5. Restart Traffic Edge.

IMPORTANT On Linux systems, you must restart Traffic Edge with the `start_traffic_server` command if you increase the RAM cache size to one GB or above; refer to *Starting Traffic Edge, on page 25*.

Inspecting the Cache

Traffic Edge provides the Cache Inspector utility, which enables you to view, delete, and invalidate URLs in the cache (HTTP only).

IMPORTANT The Cache Inspector utility is a powerful tool that is capable of deleting all the objects in your cache. Make sure that only authorized administrators are allowed to access Traffic Manager to use this utility. You can control which hosts have access to Traffic Manager in the `mgmt_allow.config` file; refer to *Controlling Host Access to Traffic Manager, on page 207*.

Accessing the Cache Inspector Utility

To access the Cache Inspector utility, use the following procedure.

▼ **To access the Cache Inspector utility:**

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Add the following variable at the end of the file:

   ```
   CONFIG proxy.config.http_ui_enabled INT 1
   ```

3. From the Traffic Edge `bin` directory, enter the following command to reread the configuration file:

   ```
   traffic_line -x
   ```

4. Open your web browser and configure it to use your Traffic Edge as a proxy server.
5. Type the following URL:

   ```
   http://{cache}
   ```

6. The Cache page opens; refer to *Using the Cache Page* below.

Using the Cache Page

The Cache page provides several options that let you view and delete the contents of your cache:

- Click **Lookup url** to search for a particular URL in the cache. When Traffic Edge finds the URL in the cache, it displays details about the object that corresponds to the URL (such as the header length and the number of alternates). From the display page, you can delete the URL from the cache.
- Click **Delete url** to delete a particular URL or list of URLs from the cache. Traffic Edge indicates if a delete is successful.
- Click **Regex lookup** to search for URLs that match one or more regular expressions. From the display page, you can delete the URLs listed; for example, enter the following to search for all URLs that end in `html` and are prefixed with `http://www.abc.com`:

  ```
  http://www.abc.com/.*\.html$
  ```

- Click **Regex delete** to delete all URLs that match a specified regular expression; for example, enter the following to delete all HTTP URLs that end in `html`:

  ```
  http://.*\.html$
  ```

- Click **Regex invalidate** to invalidate URLs that match a specified regular expression. When you invalidate a URL, Traffic Edge marks the object that corresponds to the URL as stale in the cache. Traffic Edge then contacts the origin server to check if the object is still fresh (revalidates) before serving it from the cache.

Note Make sure that no more than one administrator deletes and invalidates cache entries from the Cache page at any point in time. Changes made by multiple administrators at the same time can lead to unpredictable results.

CHAPTER 11

DNS Proxy Caching

Traffic Edge can function as a DNS proxy cache to offload remote Domain Name Servers and improve Traffic Edge performance.

This chapter discusses the following topics:

About DNS Proxy Caching

Typically, clients send DNS requests to a DNS server to resolve hostnames. However, DNS servers are typically overloaded or are not located close to the client; DNS lookups are slow and are often the bottleneck to fulfilling requests.

The DNS proxy caching option lets you resolve DNS requests on behalf of clients. This option offloads remote DNS servers and reduces response time for DNS lookups.

To use the DNS proxy caching option, Traffic Edge must be running in transparent mode.

The following overview illustrates how Traffic Edge serves a DNS request.

1. A client sends a DNS request. The request is intercepted by a router or L4 switch, which is configured to redirect all DNS traffic on port 53 to Traffic Edge.
2. The Traffic Edge ARM examines the DNS packet. If the DNS request is *type A* (answer), the ARM forwards the request to Traffic Edge. The ARM forwards all DNS requests that are not *type A* to the DNS server.
3. Traffic Edge checks its DNS cache to see if it has the hostname to IP address mapping for the DNS request. If the mapping is in the DNS cache, Traffic Edge sends the IP address to the client. If the mapping is not in the cache, Traffic Edge contacts the DNS server to resolve the hostname. When Traffic Edge receives the response from the DNS server, it caches the hostname to IP address mapping and sends the IP address to the client. If round-robin is used, Traffic Edge sends the entire list of IP address mappings to the client and the round-robin order is followed strictly.

Note If the hostname to IP address mapping is not in the DNS cache, Traffic Edge contacts the DNS server specified in the `/etc/resolv.conf` file on the Traffic Edge system. This might not be the same DNS server for which the DNS request was originally intended.

The DNS cache is held in memory and backed up on disk. Traffic Edge updates the data on disk every 60 seconds. The time-to-live is strictly followed with every hostname to IP address mapping.

IMPORTANT You can use the DNS proxy caching option with a Layer-4 switch or a Cisco router running WCCP 2.0 only. Traffic Edge does not support the DNS proxy caching option with WCCP 1.0.

Configuring DNS Proxy Caching

To configure Traffic Edge as a DNS proxy cache, you must perform the following tasks:

- Add a remap rule in the Traffic Edge `ipnat.conf` file.
- Enable the DNS proxy option and specify the port that Traffic Edge uses for DNS proxy traffic.

IMPORTANT You can use the DNS proxy option *only* if Traffic Edge is running in transparent mode. Refer to *Chapter 6, Transparent Proxy Caching* for information about configuring Traffic Edge to run in transparent mode.

You can configure Traffic Edge as a DNS proxy cache by setting options in Traffic Manager or by editing the `records.config` file manually. Both procedures are provided below.

▼ To configure DNS proxy caching from Traffic Manager:

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Networking** button and then click the **ARM** button.

   The **ARM** button appears under **Networking** only if ARM is enabled in the **Features** table under **My Proxy/Basic**.
3. In the **Network Address Translation (NAT)** section on the **General** tab, click the **Edit File** button.

   The configuration file editor for the `ipnat.conf` file opens so that you can add a rule to the `ipnat.conf` file.
4. Enter information in the fields provided:
   - In the **Ethernet Interface** field, enter the ethernet interface that DNS traffic uses to access the Traffic Edge machine: for example, `hme0` or `eth0`.
   - In the **Connection Type** drop-down list, select **udp**.
   - In the **Source IP** field, enter `0.0.0.0/0` to accept DNS traffic from all IP addresses.
   - In the **Source Port** field, enter the port on which DNS requests are sent to Traffic Edge. The default port is 53.
   - In the **Destination IP** field, enter the IP address of Traffic Edge.
   - In the **Destination Port** field, enter the port that Traffic Edge uses to communicate with the DNS server. The default port is 5353.
   - In the **User Protocol** drop-down list, select **dns**.
5. Click the **Add** button to add this rule to the `ipnat.conf` file and then click the **Apply** button to save the rule.
6. Click the **Close** button to close the configuration file editor.
7. On the **Configure** tab, click the **My Proxy** button and then click the **Basic** button.
8. In the **Features** table, click the **DNS Proxy** button in the **Networking** section.
9. Click the **Apply** button.
10. On the **Configure** tab, click the **Networking** button and then click the **DNS Proxy** button.

11 In the **DNS Proxy Port** field, enter the DNS proxy port. The default port is 53.

12 Click the **Apply** button.

13 Restart Traffic Edge by clicking the **Restart** button on the **Configure/My Proxy/Basic/General** tab.

▼ To configure DNS proxy caching manually:

1 In a text editor, open the `ipnat.conf` file located in the Traffic Edge `config` directory.

2 Add the following remap rule to the end of the file:

```
rdr interface 0.0.0.0/0 port 53 -> ipaddress port portnumber udp
```

`interface` is the network interface that receives DNS requests: for example, `hme0` or `eth0`.

`ipaddress` is the IP address of Traffic Edge.

`portnumber` is the port that Traffic Edge uses for DNS traffic. The default port is 5353.

3 Save and close the `ipnat.conf` file.

4 In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.

5 Edit the following configuration variables:

Variable	Description
proxy.config.dns.proxy.enabled	Set this variable to 1 to enable the DNS proxy caching option.
proxy.config.dns.proxy_port	Set this variable to specify the port that Traffic Edge uses for DNS traffic. The default port is 53.

6 Save and close the `records.config` file.

7 Restart Traffic Edge.

CHAPTER 12 Monitoring Traffic

Traffic Edge provides several options for monitoring system performance and analyzing network traffic.

This chapter discusses the following topics:

Traffic Edge Monitoring Tools

Traffic Edge provides the following tools to monitor system performance and analyze network traffic:

- The Traffic Manager UI provides statistics that show Traffic Edge performance and network traffic information; refer to *Viewing Statistics from Traffic Manager, on page 180*.
- The Traffic Manager UI presents alarms that signal any detected failure conditions; refer to *Working with Traffic Manager Alarms, on page 186*.
- The Traffic Line command-line interface provides an alternative method of viewing Traffic Edge performance and network traffic information; refer to *Viewing Statistics from Traffic Line, on page 188*.
- The Traffic Shell command-line tool provides yet another alternative method of viewing Traffic Edge performance and network traffic information; refer to *Starting Traffic Shell, on page 29*.
- The MRTG (Multi Router Traffic Grapher) tool provides a variety of graphs that show historical Traffic Edge performance and network traffic information; refer to *Using MRTG, on page 188*.
- SNMP (Simple Network Management Protocol) support lets you monitor and manage Traffic Edge through SNMP network management facilities; refer to *Using SNMP, on page 189*.

Viewing Statistics from Traffic Manager

You can use Traffic Manager to collect and interpret statistics about Traffic Edge performance and web traffic. You view statistics using Traffic Manager Monitor mode.

Starting Traffic Manager Monitor Mode

▼ **To start Traffic Manager Monitor mode:**

1 Open your web browser.

Traffic Manager requires Java and JavaScript; be sure to enable Java and JavaScript in your browser.

2 Type one of the following locations in your browser:

Standard `http://`*`nodename`*`:`*`adminport`*

SSL `https://`*`nodename`*`:`*`adminport`*

`nodename` is the name of the Traffic Edge node and `adminport` is the number assigned to the Traffic Manager port.

Use the `https` command to reach Traffic Manager only if you have restricted access to Traffic Manager via SSL connections; otherwise, use the standard `http` command.

3 If necessary, log on to Traffic Edge with the administrator ID and password, or use your user account.

The administrator ID and password are set during Traffic Edge installation. You can change the ID and password, as well as create and modify user accounts; refer to *Controlling Access to Traffic Manager, on page 206*.

Traffic Manager displays the **Monitor** tab (shown in *Figure 29*).

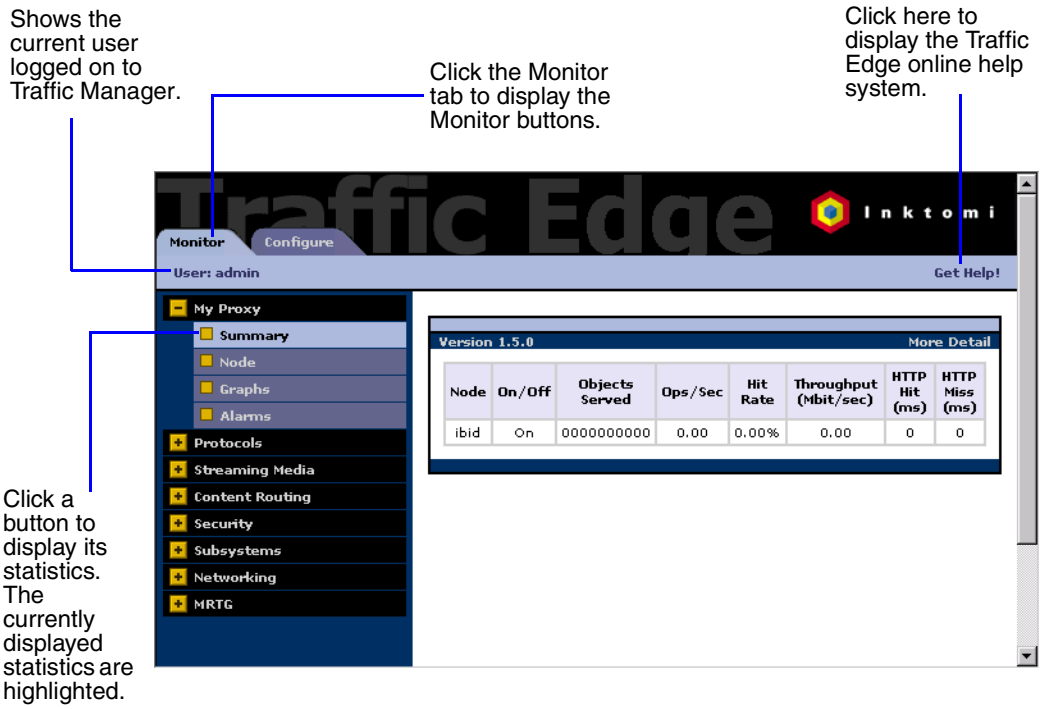


Figure 29 Traffic Manager Monitor tab

Using Monitor Mode

In Monitor mode, Traffic Manager displays a series of buttons on the left of the display. Each button represents a group of statistics. Click on a button to view its statistics. Each button is described briefly below.

All the statistics displayed in Monitor mode are described in detail in *Appendix A, Traffic Manager Statistics*.

The My Proxy Button

Click the **My Proxy** button to see statistics about your Traffic Edge proxy.

- Click the **Summary** button to see a concise view of your Traffic Edge system, displaying all cluster nodes by name and tracking essential statistics for each node. If you want to display detailed information about a particular node in a cluster, click the node's name in the **Summary** table and then click one of the other buttons on the **Monitor** tab.

  Figure 30 shows the Summary statistics.

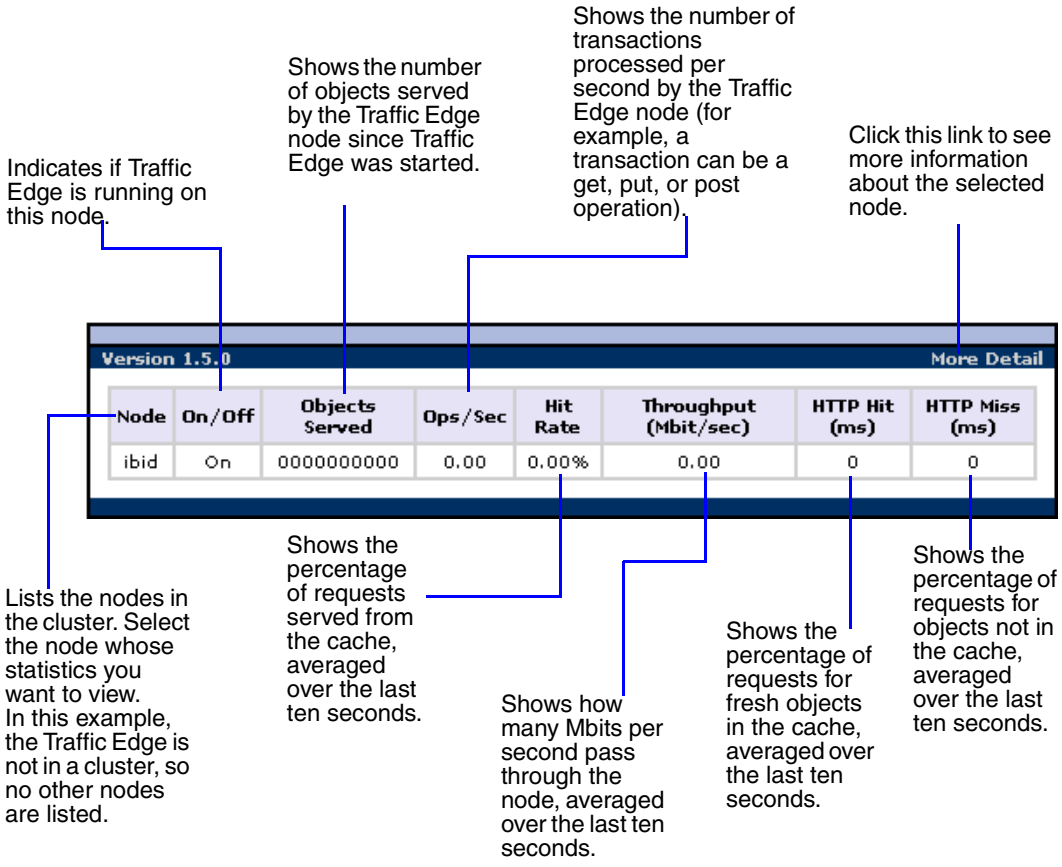


Figure 30 Summary statistics

- Click the **Node** button to see information about the selected Traffic Edge node. You can see if the node is active or inactive, the date and time that the `traffic_server` process was started, cache performance information (the document hit rate, the bandwidth savings, and what percentage of the cache is currently free), the number of client and server connections currently open, and the number of transfers currently in progress. You can also see name resolution information, such as the host database hit rate and the number of DNS lookups per second.

  If the node is part of a cluster, two sets of statistics are shown: information about the single node and information showing an average value for all the nodes in the cluster.

  Click the name of a statistic to display the information in graphical format.

- Click the **Graphs** button to view the same statistics displayed on the **Node** page (cache performance, current connections and transfers, network, and name resolution) in graphical format. You can display multiple statistics in one graph.

  To display a particular statistic in graphical format, click the box next to the name of the graph and then click the **Graph** button. To display multiple statistics in one graph, click the box next to the name of each graph you want to display and then click the **Graph** button.

- Click the **Alarms** button to view the alarms that Traffic Edge has signaled; refer to *Working with Traffic Manager Alarms, on page 186*.

The Protocols Button

The **Protocols** button provides information about HTTP and FTP transactions.

- Click the **HTTP** button to see information about HTTP transactions and speeds (such as cache misses, cache hits, connection errors, aborted transactions) and client and server connection information. You can also see information about FTP requests from HTTP clients, such as the number of open FTP server connections, the number of successful and unsuccessful PASV and PORT connections, and the number of cache lookups, hits, and misses.

- Click the **FTP** button to see information about FTP requests from FTP clients; for example, you can see the number of open FTP client and FTP server connections, the number of file hits and misses, change directory hits and misses, and list directory hits and misses.

  The **FTP** button displays only if you have enabled the FTP option.

The Streaming Media Button

The **Streaming Media** button provides information about QuickTime, Real Networks, and Windows Media transactions.

The **Streaming Media** button displays only if you have installed Traffic Edge Media Edition.

- Click the **QuickTime** button to see information about QuickTime transactions such as, the number of live streams processed, the number of open client and server connections, and the number of request and response bytes.

  The **QuickTime** button displays only if you have enabled the QuickTime option.

- Click the **Real Networks** button to see information about Real Networks transactions such as, the number of open client and server connections for on-demand and live requests, the number of on-demand and live requests since installation, and the number of request and response bytes.

  The **Real Networks** button displays only if you have enabled the Real Networks option.

- Click the **Windows Media** button to see information about Windows Media transactions such as, the number of on-demand and live streams processed, the number of open client and server connections, and the number of request and response bytes. The statistics show information for MMS-over-TCP, MMS-over-UDP, HTTP, and multicast transactions.

  The **Windows Media** button displays only if you have enabled the Windows Media option.

The Content Routing Button

Click the **Content Routing** button to see ICP statistics that include information about queries originating from the Traffic Edge node and from ICP peers (parents and siblings).

The Security Button

The **Security** button provides information about ARM, proxy authentication, and SOCKS server connections:

- Click the **ARM Security** button to see the number of dropped TCP and UDP connections.
- Click the **LDAP** button to see the number of LDAP hits and misses in the Traffic Edge authentication cache and the number of LDAP authentication server errors and unsuccessful authentication attempts.

  The **LDAP** button displays only if you have enabled the LDAP option.
- Click the **NTLM** button to see the number of NTLM hits and misses in the authentication cache, and the number of NTLM authentication server errors and unsuccessful authentication attempts.

  The **NTLM** button displays only if you have enabled the NTLM option.
- Click the **SOCKS** button to see the number of successful and unsuccessful connections to the SOCKS server and the number of connections currently in progress.

  The **SOCKS** button displays only if you have enabled the SOCKS option.

The Subsystems Button

The **Subsystems** button provides information about the Traffic Edge cache, clusters, and event logging:

- Click the **Cache** button to see information about the Traffic Edge cache. You can see how much space in the cache is currently being used, the total size of the cache in gigabytes, the total size of the RAM cache in bytes, the number of RAM cache hits and misses, and the number of cache lookups, object reads, writes, updates, and removes.
- Click the **Clustering** button to see the number of nodes in the cluster, the total number of cluster operations, the number of bytes read and written to all the nodes in the cluster, and the current number of open connections in the cluster.
- Click the **Logging** button to see the number of log files currently open, the amount of space currently being used for log files, the number of access events and error events logged, and the number of access events skipped.

The Networking Button

The **Networking** button provides information about system network configuration, the Traffic Edge ARM, WCCP routers, DNS proxy, domain name resolution, and virtual IP addressing.

- Click the **System** button to see system network configuration, which includes the hostname assigned to the Traffic Edge machine and the default gateway, search domain, and DNS servers that the Traffic Edge machine uses.
- Click the **ARM** button to see information about Network Address Translation and dynamic bypass.

  The **ARM** button displays only if you have enabled ARM.
- Click the **WCCP** button to see WCCP Version 1.0 or 2.0 statistics that include information about the routers being used, the number of active nodes, the leader's IP address, and whether WCCP is currently enabled on the Traffic Edge node.

  The **WCCP** button displays only if you have enabled WCCP.
- Click the **DNS Proxy** button to see the total number of DNS requests served by Traffic Edge, and the number of cache hits and misses.

  The **DNS Proxy** button displays only if you have enabled the DNS Proxy option.
- Click the **DNS Resolver** button to see the total number of lookups and hits in the host database, and the average lookup time, the total number of lookups, and the number of successful lookups in the DNS server.
- Click the **Virtual IP Address** button to see the current virtual IP address mappings.

  The **Virtual IP Address** button displays only if you have enabled the Virtual IP option.

The MRTG Button

Displays MRTG graphs; refer to *Using MRTG, on page 188*.

Working with Traffic Manager Alarms

Traffic Edge signals an alarm when it detects a problem; for example, if the space allocated to event logs is full or if Traffic Edge cannot write to a configuration file.

Traffic Edge displays the alarms it detects on the Traffic Manager **Monitor** tab. To view the current alarms, click the **My Proxy** button on the **Monitor** tab and then click the **Alarms** button.

Figure 31 shows the **Alarms** area in Traffic Manager.

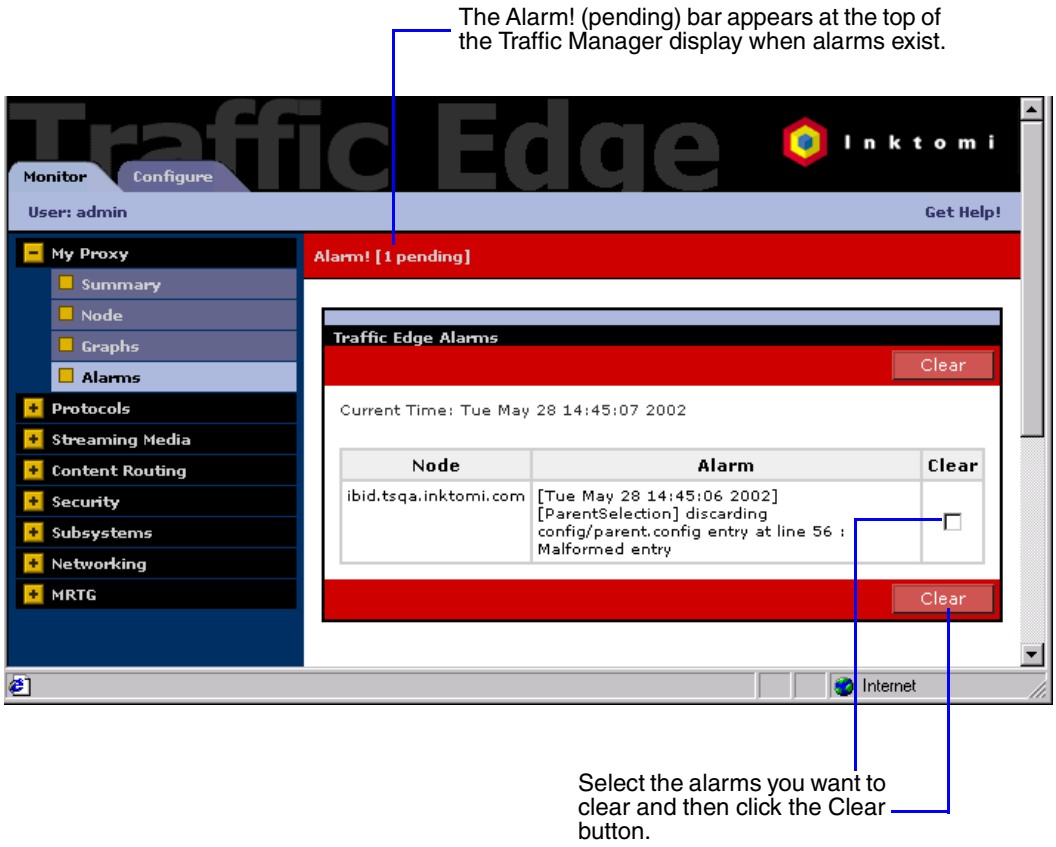


Figure 31 Alarms in Traffic Manager

Clearing Alarms

After you have read an alarm message, you can click the **Clear** button in the alarm message window to dismiss the alarm. *Traffic Edge Alarm Messages, on page 467*, provides a description of some of the alarm messages that Traffic Edge provides.

IMPORTANT Clicking the **Clear** button only dismisses alarm messages; it does not actually resolve the cause of the alarms.

Configuring Traffic Edge to Email Alarms

You can configure Traffic Edge to send an email to a specific address whenever an alarm occurs. You can do this from Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ **To specify an email address for alarms from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. Click the **Configure** tab.

   The **General** tab of the **My Proxy**/**Basic** button displays.
3. In the **Alarm E-Mail** field, enter the email address to which you want to send alarms.
4. Click the **Apply** button.

▼ **To specify an email address for alarms manually:**

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Set the variable `proxy.config.alarm_email` to the email address to which you want to send alarms.
3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

Using a Script File for Alarms

Alarm messages are built into Traffic Edge: you cannot change them. However, you can write a script file to execute certain actions when an alarm is signaled.

Traffic Edge provides a sample script file named `example_alarm_bin.sh` (UNIX) or `example_alarm_bin.bat` (Windows) in the `bin` directory. You can modify the file to suit your needs.

Viewing Statistics from Traffic Line

As an alternative to using Traffic Manager, you can use the Traffic Line command-line interface to view statistics about Traffic Edge performance and web traffic. Traffic Line provides a quick way of viewing Traffic Edge statistics if you do not have access to a browser or if you prefer to use a UNIX shell-like command interface.

In addition to viewing statistics, you can also configure, stop, and restart the Traffic Edge system; refer to *Configuring Traffic Edge Using Traffic Line, on page 196*, and *Appendix C, Traffic Line Commands*.

You can view specific information about a Traffic Edge node or cluster by specifying the variable that corresponds to the statistic you want to see.

▼ **To view a statistic:**

1 In UNIX, log on to a Traffic Edge node as the Traffic Edge administrator and then navigate to the Traffic Edge `bin` directory.

In Windows, open a Command Prompt window and then navigate to the Traffic Edge `bin` directory.

2 Enter the following command:

```
traffic_line -r variable
```

`variable` is the variable that represents the information you want to view. For a list of the variables you can specify, refer to *Traffic Line Variables, on page 353*; for example, the following command displays the document hit rate for the Traffic Edge node:

```
traffic_line -r proxy.node.http.cache_hit_ratio
```

In UNIX, if the Traffic Edge `bin` directory is not in your path, prepend the Traffic Line command with `./` (for example, `./traffic_line -r variable`).

Using MRTG

The MRTG (Multi Router Traffic Grapher) tool lets you monitor Traffic Edge performance and analyze network traffic. MRTG provides a variety of graphs that show information about virtual memory usage, client connections, cache hit and miss rates, and so on. The information provided is recorded from the time that Traffic Edge was started. MRTG uses five-minute intervals to formulate the statistics and provides useful historical information.

You access MRTG from the Traffic Manager **Monitor** tab.

IMPORTANT To run MRTG in UNIX, you must have Perl Version 5.005 or later installed on your Traffic Edge system. To run MRTG in Windows, you must have Windows Services for UNIX (SFU) 2.0 or later installed on your Traffic Edge system.

▼ **To access MRTG:**

1 From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.

2 If your Traffic Edge node is in a cluster, select the Traffic Edge node whose statistics you want to view from the **My Proxy**/**Summary** display on the **Monitor** tab.

3 On the **Monitor** tab, click the **MRTG** button.

4 Click the **Overview** button to see a subset of the graphs available in MRTG.

Click the **Daily** button to see Traffic Edge statistics for the current day.

Click the **Weekly** button to see Traffic Edge statistics for the current week.

Click the **Monthly** button to see Traffic Edge statistics for the current month.

Click the **Yearly** button to see Traffic Edge statistics for the current year.

5 Wait at least fifteen minutes before looking at the graphs. It takes several five-minute sample intervals for MRTG to initialize statistics.

Using SNMP

The Simple Network Management Protocol (SNMP) is a standard protocol used for network management. SNMP agents collect and store management information in Management Information Bases (MIBs), and SNMP managers can probe the agents for this information. In addition, SNMP agents can send alarms and alerts called *SNMP traps* to the SNMP manager to warn of any problems.

The Traffic Edge SNMP agent supports access to two MIBs: MIB-2 (a standard MIB) and the Inktomi Traffic Edge MIB. Descriptions of the Traffic Edge MIB variables are provided in the `inktomi-ts-mib.my` file in the Traffic Edge `config/mibs` directory. The Traffic Edge MIB contains both node-specific and cluster-wide information.

To use SNMP on your Traffic Edge system, you need to perform the following tasks:

- Control MIB access to specific hosts; refer to *Controlling MIB Access*, below.
- Configure Traffic Edge to send SNMP traps; refer to *Configuring SNMP Trap Destinations* below.
- Enable the Traffic Edge SNMP agent; refer to *Enabling SNMP, on page 190*.

Controlling MIB Access

By default, read-only access to the Traffic Edge MIBs is granted to any host that makes SNMP requests using the community string `public`. Inktomi recommends that you configure your Traffic Edge system to control MIB access so that only certain hosts can access SNMP information.

To configure Traffic Edge to control MIB access, edit the `snmpd.cnf` file located in the Traffic Edge `config` directory; refer to *snmpd.cnf, on page 449*.

Configuring SNMP Trap Destinations

To configure SNMP trap destinations, edit the `snmpd.cnf` file located in the Traffic Edge `config` directory; refer to *snmpd.cnf, on page 449*.

Enabling SNMP

You must enable the Traffic Edge SNMP agent so that SNMP managers can access the MIBs and gather information. You can enable the SNMP agent either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ To enable the SNMP agent from Traffic Manager:

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. Click the **Configure** tab.

   The **General** tab of the **My Proxy**/**Basic** button displays.
3. Click the **SNMP On** button in the **Features** table.
4. Click the **Apply** button.

▼ To enable the SNMP agent manually:

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Edit the following variable:

Variable	Description
proxy.config.snmp.master_agent_enabled	Set this variable to 1 to enable SNMP on the Traffic Edge node

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

CHAPTER 13

Configuring Traffic Edge

Traffic Edge provides several options for configuring the system.

This chapter discusses the following topics:

Configuring Traffic Edge Using Traffic Manager

You can use Traffic Manager to view and change your Traffic Edge configuration. You set configuration options using Traffic Manager Configure mode.

Note Certain Traffic Edge configuration options can only be changed by editing configuration variables in the `records.config` file, in Traffic Line, or in Traffic Shell; refer to *Configuring Traffic Edge Using Traffic Line, on page 196*, *Configuring Traffic Edge Using Configuration Files, on page 197*, and *Starting Traffic Shell, on page 29*.

Starting Traffic Manager Configure Mode

▼ **To start Traffic Manager Configure mode:**

1. Open your web browser.

   Traffic Manager requires Java and JavaScript; be sure to enable Java and JavaScript in your browser.

2. Type one of the following locations in your browser:

Standard `http://nodename:adminport`

SSL `https://nodename:adminport`

`nodename` is the name of the Traffic Edge node and `adminport` is the number assigned to the Traffic Manager port.

Use the `https` command to access Traffic Manager only if you have restricted access to Traffic Manager via SSL connections; otherwise, use the standard `http` command.

3 If necessary, log on to Traffic Edge with the administrator ID and password or with your user account.

The administrator ID and password are set during Traffic Edge installation. You can change the ID and password, as well as create and modify user accounts. For more information, refer to *Controlling Access to Traffic Manager, on page 206*.

Traffic Manager starts by default in Monitor mode.

4 Click the **Configure** tab to display the Configure mode buttons (shown in *Figure 32*).

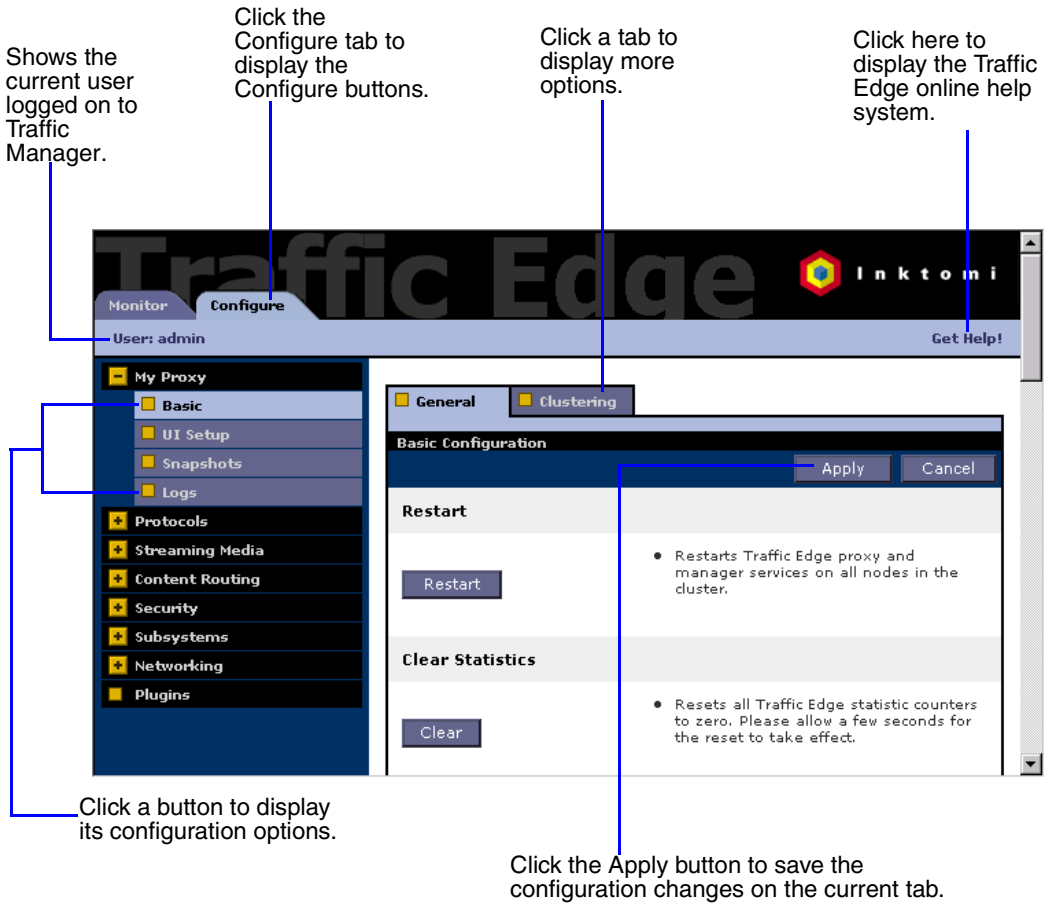


Figure 32 Traffic Manager Configure tab

Using Configure Mode

In Configure mode, Traffic Manager displays a series of buttons on the left side of the display. Each button represents a group of configuration options. Each button is described briefly below.

All the configuration options available in Configure mode are described in *Appendix B, Traffic Manager Configuration Options*.

The My Proxy Button

The **My Proxy** button contains the Basic, UI Setup, Snapshots, and Logs configuration groups:

- Click **Basic** to restart the Traffic Edge proxy and manager services (you need to restart after changing certain configuration options), reset the statistics, identify the name of the Traffic Edge node, set alarm email, enable or disable Traffic Edge features (such as proxy authentication, ARM, WCCP, and so on) and set cluster options.
- Click **UI Setup** to identify and change the port on which browsers can connect to Traffic Manager, enable SSL connections to Traffic Manager, specify how often Traffic Manager refreshes the statistics on the **Monitor** tab, and configure access control lists, administrator accounts, and user accounts to secure Traffic Manager access.
- Click **Snapshots** to take and restore configuration snapshots.
- Click **Logs** to display, delete, or copy a selected log file to the local filesystem.

The Protocols Button

The **Protocols** button contains the HTTP, HTTP Responses, HTTP Scheduled Update, and FTP configuration groups:

- Click **HTTP** to configure HTTP caching and to tune HTTP timeouts.
- Click **HTTP Responses** to specify which HTTP responses are sent to clients when Traffic Edge detects an HTTP problem with a particular client transaction (such as unavailable origin servers, authentication requirements, and protocol errors).
- Click **HTTP Scheduled Update** to configure Traffic Edge to load specific objects into the cache at scheduled times.
- Click **FTP** to configure FTP caching and to tune FTP timeouts. The FTP options affect requests that originate from FTP clients only. You can configure options that affect FTP requests originating from HTTP clients from the HTTP group.

  The **FTP** button displays only if you have enabled the FTP option.

The Streaming Media Button

The **Streaming Media** button contains the Shared Settings, QuickTime, Real Networks, and Windows Media configuration groups. The **Streaming Media** button displays only if you have installed Traffic Edge Media Edition.

- Click the **Shared Settings** button to configure the RTSP proxy port setting, which is used for QuickTime and Real Networks requests.
- Click the **QuickTime** button to configure QuickTime MediaBridge options.

  The **QuickTime** button displays only if you have enabled the QuickTime option.
- Click the **Real Networks** button to configure the RealProxy restart limit.

  The **Real Networks** button displays only if you have enabled the Real Networks option.

- Click the **Windows Media** button to configure Windows Media settings such as, the Windows Media proxy port, ASX rewrite, multicast, media push, and MediaBridge.

  The **Windows Media** button displays only if you have enabled the Windows Media option.

The Content Routing Button

The **Content Routing** button contains the Hierarchies, Reverse Proxy, Mapping and Redirection, and Browser Auto Config groups:

- Click **Hierarchies** to configure parent caching and ICP peer options.
- Click **Reverse Proxy** to enable the reverse proxy option for HTTP, streaming media, and FTP protocols.
- Click **Mapping and Redirection** to set URL remapping rules and FTP remapping rules.
- Click **Browser Auto-Config** to identify the port used to download browser autoconfiguration files, and to set PAC and WPAD options.

The Security Button

The **Security** button contains the Connection Control, Access Control, SSL Termination, and SOCKS configuration groups:

- Click **Connection Control** to specify which hosts are allowed to communicate with the Traffic Edge machine and which clients are allowed to access the proxy cache.
- Click **Access Control** to set filtering rules and set proxy authentication options (LDAP, RADIUS, and NTLM).
- Click **SSL Termination** to set SSL termination options. SSL termination enables you to secure connections in reverse proxy mode between a client and Traffic Edge, and/or Traffic Edge and an origin server.

  The **SSL Termination** button displays only if you have enabled the SSL Termination **option**.

- Click **SOCKS** to configure Traffic Edge to use a SOCKS firewall.

  The **SOCKS** button displays only if you have enabled the SOCKS option.

The Subsystems Button

The **Subsystems** button contains the Cache and Logging configuration groups:

- Click **Cache** to enable or disable cache pinning, configure the RAM cache size, specify the maximum size of objects allowed in the cache, and partition your cache according to protocol and origin servers.
- Click **Logging** to enable or disable event logging and set logging configuration options.

The Networking Button

The **Networking** button contains the System, Connection Management, ARM, WCCP, DNS Proxy, DNS Resolver, and Virtual IP configuration groups:

- Click **System** to change your network configuration (such as the hostname of the machine running Traffic Edge, and the default gateway and DNS servers that the machine uses).
- Click **Connection Management** to specify the maximum number of connections Traffic Edge can accept. For transparent proxy caching, you can specify the maximum number of client connections allowed before Traffic Edge starts forwarding incoming requests directly to the origin server.
- Click **ARM** to set redirection rules that specify how incoming packets are readdressed in transparent mode. You can also set dynamic and static bypass rules.

  The **ARM** button displays only if you have enabled the ARM.
- Click **WCCP** to set WCCP configuration settings.

  The **WCCP** button displays only if you have enabled WCCP.
- Click **DNS Proxy** to specify the DNS proxy port.

  The **DNS Proxy** button displays only if you have enabled the DNS Proxy option.
- Click **DNS Resolver** to enable or disable local domain expansion, tune host database timeouts, and configure Split DNS options.
- Click **Virtual IP** to enable or disable virtual IP failover and specify the virtual IP addresses managed by the Traffic Edge node.

  The **Virtual IP** button displays only if you have enabled the Virtual IP option.

The Plugins Button

Click the **Plugins** button to list the plugins currently running on your Traffic Edge that are configurable from Traffic Manager. A plugin is a program that extends the functionality of Traffic Edge; for example, you can run plugins to blacklist web servers, filter web content, authenticate users, and transform data.

Configuring Traffic Edge Using Traffic Line

As an alternative to using Traffic Manager, you can use the Traffic Line command-line interface to view and change your Traffic Edge configuration. Traffic Line provides a quick way of configuring your system if you do not have access to a browser or if you prefer to use a UNIX shell-like command interface.

▼ **To view or change configuration options in Traffic Line:**

1. In UNIX, log on to a Traffic Edge node as the Traffic Edge administrator and then navigate to the Traffic Edge `bin` directory.

   In Windows, open a Command Prompt window and then navigate to the Traffic Edge `bin` directory.

2. To view a configuration setting, enter the following command:

   ```
   traffic_line -r var
   ```

   `var` is the variable associated with the configuration option (for a list of the variables, refer to *Configuration Variables, on page 398*).

3. To change the value of a configuration setting, enter the following command:

   ```
   traffic_line -s var -v value
   ```

   `var` is the variable associated with the configuration option (for a list of the variables, refer to *Configuration Variables, on page 398*) and *`value`* is the value you want to use; for example, to change the FTP inactivity timeout option to 200 seconds, enter the following command at the prompt and press Return:

   ```
   traffic_line -s proxy.config.ftp.control_connection_timeout -v 200
   ```

   In UNIX, if the Traffic Edge `bin` directory is not in your path, prepend the Traffic Line command with `./` (for example, `./traffic_line -r variable`).

As an alternative to Traffic Line, you can use Traffic Shell to configure Traffic Edge; refer to *Starting Traffic Shell, on page 29*.

Configuring Traffic Edge Using Configuration Files

As an alternative to using Traffic Manager, Traffic Line, or Traffic Shell, you can change Traffic Edge configuration options by manually editing specific variables in the `records.config` file.

The `records.config` file is located in the Traffic Edge `config` directory. To edit the variables, open the file in a text editor (such as `vi` or `emacs`) and change the variable value.

After you modify the `records.config` file, Traffic Edge has to reread the configuration files; from the Traffic Edge `bin` directory, enter the Traffic Line command `traffic_line -x`. In some cases, you have to restart Traffic Edge to apply the configuration changes.

The following is a sample portion of the `records.config` file:

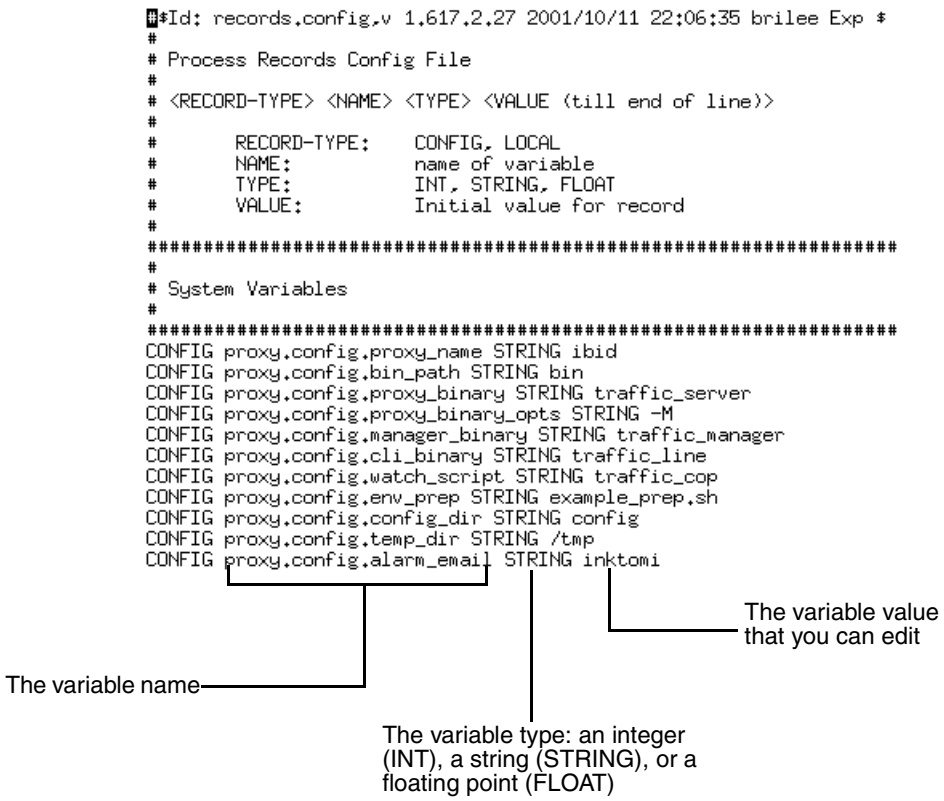


Figure 33 Sample records.config file

In addition to the `records.config` file, Traffic Edge provides other configuration files that are used to configure specific features. All the configuration files that you can edit manually are described in *Appendix E, Configuration Files*.

Saving and Restoring Traffic Edge Configurations

Traffic Edge provides a configuration snapshot feature that lets you save all the current configuration settings and restore them when necessary. Traffic Edge can store configuration snapshots on the node on which they are taken, on an FTP server, or on a floppy disk. Traffic Edge restores a configuration snapshot on all the nodes in the cluster.

Inktomi recommends that you take a configuration snapshot before performing system maintenance or attempting to tune system performance. Taking a configuration snapshot only takes a few seconds and it can save you hours of correcting configuration mistakes.

This section describes how to perform the following tasks:

- Take a snapshot of the current configuration; refer to *Taking Configuration Snapshots*, below.
- Restore previously taken configuration snapshots; refer to *Restoring Configuration Snapshots, on page 199*.
- Delete configuration snapshots stored on the Traffic Edge node; refer to *Deleting Configuration Snapshots, on page 201*.

You can save, restore and delete configuration snapshots from an FTP server or floppy disk on Linux systems only.

Taking Configuration Snapshots

You can save all the current configuration settings on your Traffic Edge system by using Traffic Manager. You can save the current configuration settings to a location on the local system, to an FTP server, or to a floppy disk.

▼ **To take a configuration snapshot and save it on the local system:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. Click the **Configure** tab.

   The **General** tab of the **My Proxy**/**Basic** button displays.
3. Click the **Snapshots** button.
4. The **Change Snapshot Directory** field on the **File System** tab displays the name of the directory in which Traffic Edge saves configuration snapshots. The default location is **config/snapshots** in the directory where Traffic Edge is installed. To change the directory, enter the full path in the **Change Snapshot Directory** field. If you enter a relative path, Traffic Edge assumes the directory is located in its `config` directory; for example, `inktomi/traffic_server/config`.
5. In the **Save Snapshot** field, type the name you want to use for the current configuration.
6. Click the **Apply** button.

   Traffic Edge takes the configuration snapshot and saves it on the local system.

▼ **To take a configuration snapshot and save it on an FTP server:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. Click the **Configure** tab.

   The **General** tab of the **My Proxy/Basic** button displays.
3. Click the **Snapshots** button.
4. Click the **FTP Server** tab.
5. In the fields provided, enter the FTP server name, the login and password, and the remote directory in which the FTP server stores configuration snapshots.
6. Click the **Apply** button.

   After you have successfully logged on to the FTP server, the **FTP Server** tab displays additional fields.
7. In the **Save Snapshot to FTP Server** field, enter the name of the configuration snapshot you want to take.
8. Click the **Apply** button.

   Traffic Edge takes the configuration snapshot and saves it in the specified remote directory on the FTP server.

▼ **To take a configuration snapshot and save it on a floppy disk:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. Click the **Configure** tab.

   The **General** tab of the **My Proxy/Basic** button displays.
3. Click the **Snapshots** button.
4. Click the **Floppy Disk** tab.

   Traffic Edge lists the floppy disk drives on the system.
5. From the **Select Floppy Drive** field, select the floppy disk drive on which you want to save a configuration snapshot.

   The **Floppy Disk** tab displays additional fields.
6. In the **Save Snapshot** field, enter the name of the configuration snapshot you want to take.
7. Click the **Apply** button.

   Traffic Edge takes the configuration snapshot and saves it on the floppy disk.

Restoring Configuration Snapshots

You can restore a configuration snapshot from the local Traffic Edge node, from an FTP server or from a floppy disk.

If you are running a cluster of Traffic Edges, the configuration is restored to all the nodes in the cluster.

▼ **To restore a configuration snapshot stored on the local node:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. Click the **Configure** tab.

   The **General** tab of the **My Proxy**/**Basic** button displays.
3. Click the **Snapshots** button.
4. From the **Restore/Delete Snapshot** drop-down list on the **File System** tab, select the configuration snapshot that you want to restore.
5. Click the **Restore Snapshot from "*directory_name*" Directory** box.
6. Click the **Apply** button.

   The Traffic Edge system or cluster uses the restored configuration.

▼ **To restore a configuration snapshot from an FTP server:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. Click the **Configure** tab.

   The **General** tab of the **My Proxy**/**Basic** button displays.
3. Click the **Snapshots** button.
4. Click the **FTP Server** tab.
5. In the fields provided, enter the FTP server name, the login and password, and the remote directory in which the FTP server stores configuration snapshots.
6. Click the **Apply** button.

   After you have successfully logged on to the FTP server, the **FTP Server** tab displays additional fields.
7. In the **Restore Snapshot** drop-down list, select the configuration snapshot that you want to restore.
8. Click the **Apply** button.

   The Traffic Edge system or cluster uses the restored configuration.

▼ **To restore a configuration snapshot from floppy disk:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. Click the **Configure** tab.

   The **General** tab of the **My Proxy**/**Basic** button displays.
3. Click the **Snapshots** button.
4. Click the **Floppy Disk** tab.

   Traffic Edge lists the floppy disk drives on the system.
5. From the **Select Floppy Drive** field, select the floppy disk drive from which you want to restore a configuration snapshot.

   The **Floppy Disk** tab displays additional fields.

6 In the **Restore Snapshot** drop-down list, select the configuration snapshot that you want to restore.

7 Click the **Apply** button.

The Traffic Edge system or cluster uses the restored configuration.

Deleting Configuration Snapshots

You can delete any of the configuration snapshots stored on the local Traffic Edge node by using Traffic Manager.

▼ **To delete a configuration snapshot:**

1 From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.

2 Click the **Configure** tab.

The **General** tab of the **My Proxy**/**Basic** button displays.

3 Click the **Snapshots** button.

4 From the **Restore/Delete a Snapshot** drop-down list on the **File System** tab, select the configuration snapshot you want to delete.

5 Click the **Delete Snapshot from "*directory_name*" directory** box.

6 Click the **Apply** button.

Traffic Edge deletes the configuration snapshot.

CHAPTER 14 Security Options

Traffic Edge provides a number of security features.

This chapter discusses the following topics:

Controlling Client Access to the Proxy Cache

You can configure Traffic Edge to allow only certain clients to use the proxy cache either by using Traffic Manager or by editing a configuration file manually.

▼ **To specify the clients allowed to use the proxy cache from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Security** button and then click the **Connection Control** button.
3. In the **Access Control** area on the **Proxy Access** tab, click the **Edit File** button.

   The configuration file editor for the `ip_allow.config` file opens.
4. Enter information in the fields provided and then click the **Add** button. All the fields are described in *Appendix B, Traffic Manager Configuration Options*.
5. Click the **Apply** button to save the information and then click the **Close** button to exit the configuration file editor.

▼ **To specify the clients allowed to use the proxy cache manually:**

1. In a text editor, open the `ip_allow.config` file located in the Traffic Edge `config` directory.
2. Add a line in the file for each IP address or range of IP addresses allowed to access Traffic Edge; refer to *ip_allow.config, on page 383*.

3 Save and close the `ip_allow.config` file.

4 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

5 Run the command `traffic_line -x` to apply the configuration changes.

Note If an unauthorized client tries to access Traffic Edge, a message displays in their browser, indicating that the requested content cannot be obtained; for example, in Netscape Version 4.7, the message *The document contained no data* appears in the browser window. In Internet Explorer Version 5.0, the message *The page cannot be displayed* appears in the browser window.

Controlling Host Access to the Traffic Edge Machine

For security reasons, you might want to restrict the type of communication possible with machines running Traffic Edge. Using the Traffic Edge ARM security option, you can create an access control list that is used to either allow or deny other hosts from communicating with the Traffic Edge machine on specific ports. This *firewall* prevents potentially malicious packets from disrupting the operation of the machine.

When the ARM security option is enabled, the Traffic Edge ARM examines UDP and TCP packets as they arrive at the Traffic Edge machine and matches them against the access control list that you specify in a configuration file. The ARM checks all UDP packets (since UDP communication is, by definition, connectionless) and looks at the first TCP packet initiating the session against the configuration file access control list. Acceptable packets using either protocol are then passed up the network stack. Only incoming UDP and TCP packets are affected. This means that it is always possible to initiate TCP and UDP connections from the Traffic Edge, regardless of the access control list configured.

To use the ARM security feature, you must perform the following procedures in the order listed:

- Edit the `arm_security.config` file to open specific ports and define the hosts that are allowed to communicate with the Traffic Edge machine.
- Enable the ARM security option.

IMPORTANT By default, when you enable the ARM security option, all ports on the Traffic Edge machine are closed, except for the Traffic Edge ports, DNS service ports, and ssh port 22. Before you enable the ARM security option, ensure that you have either console access to the Traffic Edge machine or that you have added the appropriate rules to the `arm_security.config` file to allow `telnet` or `ssh` access for yourself.

When you enable the ARM security option, the ports you specify in the access control list remain closed even when Traffic Edge is not running.

IMPORTANT You must define ports and hosts in the `arm_security.config` file before you enable the ARM Security option so that you do not lock yourself out of the Traffic Edge machine.

You can edit the `arm_security.config` file and enable the ARM security option either by using Traffic Manager or by editing configuration files manually. Both procedures are provided below.

▼ To edit the arm_security.config file and enable ARM security from Traffic Manager:

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Security** button and then click the **Connection Control** button.
3. Click the **ARM Security** tab.
4. In the **Access Control List** area, click the **Edit File** button.

   The configuration file editor for the `arm_security.config` file opens.
5. Enter information in the fields provided and then click the **Add** button. All the fields are described in *Appendix B, Traffic Manager Configuration Options*.
6. Click the **Apply** button to save the information and then click the **Close** button to exit the configuration file editor.
7. On the **ARM Security** tab, enable the **ARM Security** option.
8. Click the **Apply** button to save your configuration.
9. Restart Traffic Edge by clicking the **Restart** button on the **Configure**/**My Proxy**/**Basic/General** tab.

▼ To edit the arm_security.config file and enable ARM security manually:

1. In a text editor, open the `arm_security.config` file located in the Traffic Edge `config` directory.
2. Add open, allow, and deny rules to define which ports you want to remain open and which hosts are allowed to communicate with Traffic Edge; refer to *arm_security.config, on page 366*.
3. Save and close the `arm_security.config` file.
4. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
5. Edit the following variable:

Variable	Description
proxy.config.arm.security_enabled	Set this variable to 1 to enable ARM security.

6. Save and close the `records.config` file.
7. Restart Traffic Edge.

Controlling Access to Traffic Manager

You can restrict access to Traffic Manager to ensure that only authenticated users can change Traffic Edge configuration options and view performance and network traffic statistics. You can:

- Set the master administrator ID and password. A user that logs on to Traffic Manager with the administrator ID has access to all Traffic Manager activities; refer to *Setting the Administrator ID and Password*, below.
- Create and maintain a list of user accounts that determines who can log on to Traffic Manager and which activities they can perform; refer to *Creating a List of User Accounts, on page 207*.
- Create an access control list of IP addresses that defines which machines can access Traffic Manager; refer to *Controlling Host Access to Traffic Manager, on page 207*.
- Use SSL for secure administration; refer to *Using SSL for Secure Administration, on page 208*.

Setting the Administrator ID and Password

During Traffic Edge installation, you assign an administrator ID and password that controls access to Traffic Manager. A user that logs on to Traffic Manager using the correct ID and password can view all the statistics on the **Monitor** tab and change any configuration options on the **Configure** tab.

You can change the administrator ID and password at any time.

▼ To change the administrator ID and password:

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **UI Setup** button under **My Proxy**.
3. Click the **Login** tab.
4. Make sure that the **Basic Authentication** option is enabled.

   When the **Basic Authentication** option is disabled, any user can access Traffic Manager unless you have set up a list of IP addresses that are denied access; refer to *Controlling Host Access to Traffic Manager, on page 207*.
5. To change the current administrator ID, type a new ID in the **Login** field of the **Administrator** section.
6. To change the current password, type the current password in the **Old Password** field. Type the new password in the **New Password** field and then retype the new password in the **New Password (Retype)** field.

   If you have forgotten the current administrator password, refer to *How do you access Traffic Manager if you forget the master administrator password?, on page 475*.
7. Click the **Apply** button.

Creating a List of User Accounts

If a single administrator ID and password for Traffic Manager is not sufficient security for your needs, you can create a list of user accounts that define who has access to Traffic Manager and which activities they can perform.

You can use user accounts in addition to using the administrator ID and password.

▼ **To create a list of user accounts:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **UI Setup** button under **My Proxy**.
3. Click the **Login** tab.
4. In the **New User** field of the **Additional Users** area, enter the name of the user allowed to access Traffic Manager.
5. In the **New Password** field, enter the password for the user and then enter the password again in the **New Password (Retype)** field.
6. Click the **Apply** button.
7. In the **Access** drop-down list of the user table, select which Traffic Manager activities the user can perform:
   - Select **No Access** to disable Traffic Manager access for the user.
   - Select **Monitor Only** to allow the user to view statistics from the **Monitor** tab only.
   - Select **Monitor and View Configuration** to allow the user to view statistics from the **Monitor** tab and to *view* configuration options from the **Configure** tab.
   - Select **Monitor and Modify Configuration** to allow the user to view statistics from the **Monitor** tab and to change configuration options from the **Configure** tab.
8. Click the **Apply** button.
9. Repeat *step 4* through *step 8* for each user allowed to access Traffic Manager.
10. Make sure the **Basic Authentication** option is enabled.

    Traffic Edge checks usernames and passwords only if this option is enabled.

Controlling Host Access to Traffic Manager

In addition to using an administrator ID and user accounts, you can control which hosts have access to Traffic Manager.

▼ **To control host access to Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **UI Setup** button under **My Proxy**.
3. Click the **Access** tab.
4. In the **Access Control** area, click the **Edit File** button.

   The configuration file editor for the `mgmt_allow.config` file opens.
5. Enter information in the fields provided and then click the **Add** button. All the fields are described in *Appendix B, Traffic Manager Configuration Options*.

6 Click the **Apply** button to save the information and then click the **Close** button to exit the configuration file editor.

▼ **To control host access to Traffic Manager manually:**

1 In a text editor, open the `mgmt_allow.config` file located in the Traffic Edge `config` directory.

By default, the file contains the following line, which allows all hosts to access Traffic Manager:

```
src_ip=0.0.0.0-255.255.255.255    action=ip_allow
```

2 Comment out the default line, as shown:

```
#src_ip=0.0.0.0-255.255.255.255 action=ip_allow
```

3 Add a line for each IP address or range of IP addresses allowed to access Traffic Manager; refer to *mgmt_allow.config, on page 393*.

4 Save and close the `mgmt_allow.config` file.

5 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

6 Run the command `traffic_line -x` to apply the configuration changes.

Using SSL for Secure Administration

Traffic Edge supports the Secure Sockets Layer protocol (SSL) to provide protection for remote administrative monitoring and configuration using Traffic Manager. SSL security provides authentication of both ends of a network connection using certificates and provides privacy using encryption.

To use SSL, you must perform the following procedures:

- Obtain an SSL certificate
- Enable the Traffic Manager SSL option
- Access Traffic Manager using the `https` command

Obtaining an SSL Certificate

You can obtain an SSL certificate from either:

- Inktomi Technical Support

  Use your websupport access account to obtain an SSL certificate. The certificate is a text file that you must install in the Traffic Edge `config` directory. Each time you connect to Traffic Manager from your browser using the SSL certificate you obtained from Inktomi, you must go through an interactive acceptance dialogue.

- A recognized certificate authority: for example, VeriSign

  Install the certificate in the Traffic Edge `config` directory. You must either rename the certificate to the default filename `private_key.pem`, or specify the name of the certificate in Traffic Manager or in the configuration file (follow the procedure in *Enabling SSL*, below).

Enabling SSL

After you have obtained an SSL certificate, you can enable SSL either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ **To enable SSL from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **UI Setup** button under **My Proxy**.
3. On the **General** tab, enable the **HTTPS** option.
4. In the **Certificate File** field, specify the filename of the SSL certificate.

   You have to change the filename only if the certificate file does not use the default name `private_key.pem`.
5. Click the **Apply** button.

▼ **To enable SSL manually:**

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Edit the following variables:

Variable	Description
proxy.config.admin.use_ssl	Set this variable to 1 to enable SSL.
proxy.config.admin.ssl_cert_file	Set this variable to specify the filename of the SSL certificate. You have to change the filename only if the certificate file does not use the default name `private_key.pem`.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

Accessing Traffic Manager Using SSL

To access Traffic Manager from your browser using SSL, use the `https` command as shown:

```
https://nodename:adminport
```

`nodename` is the hostname of the Traffic Edge node and *`adminport`* is the port number assigned to the Traffic Manager port (the default port number is 8081).

Configuring SOCKS Firewall Integration

SOCKS is commonly used as a network firewall that allows hosts behind a SOCKS server to gain full access to the Internet and prevents unauthorized access from the Internet to hosts inside the firewall.

Figure 34 illustrates how Traffic Edge integrates into a SOCKS firewall.

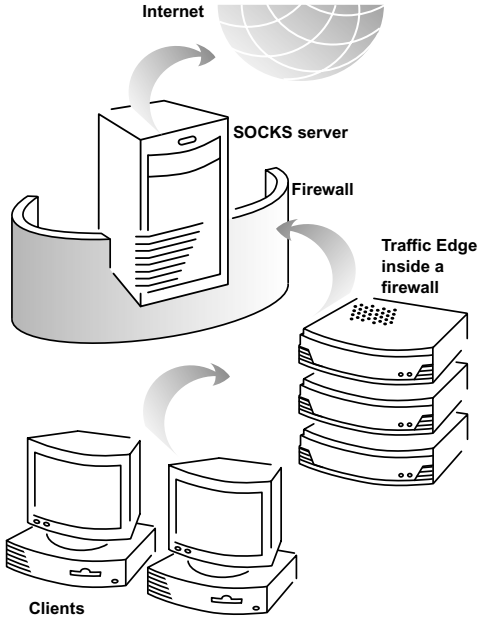


Figure 34 The Traffic Edge inside a firewall using a SOCKS server

When Traffic Edge receives a request for content that is not in the cache or is stale, it must request the content from the origin server. In a SOCKS configuration, instead of accessing the origin server directly, Traffic Edge goes through a SOCKS server. The SOCKS server authorizes communication between Traffic Edge and the origin server and then relays the data to the origin server. The origin server then sends the content back to Traffic Edge through the SOCKS server. Traffic Edge caches the content and sends it to the client.

Traffic Edge can act as a *SOCKS client*, where it receives and serves HTTP or FTP requests as usual, and as a *SOCKS proxy*, where it receives SOCKS traffic (usually on port 1080). As a SOCKS proxy, Traffic Edge detects and serves HTTP requests but forwards all other requests directly to the SOCKS server.

Note Traffic Edge does not perform authentication with the client. However Traffic Edge can perform username and password authentication with a SOCKS server running SOCKS Version 5.

To configure your Traffic Edge to use a SOCKS firewall, you perform the following tasks:

- Enable the SOCKS option.
- Specify the hostnames of your default SOCKS servers and the communication ports.
- Specify the IP addresses of any origin servers that you want Traffic Edge to access directly without going through the SOCKS server (optional).

- Specify the username and password Traffic Edge must use for authentication with a SOCKS server running Version 5 (optional).

Refer to *Configuring Traffic Edge to Use a SOCKS Firewall, on page 211*.

To use Traffic Edge as a SOCKS proxy, you must perform the following procedures *in addition* to the procedures outlined above:

- Enable the *SOCKS proxy* option.
- Specify the port used to receive SOCKS traffic.

Refer to *Setting SOCKS Proxy Options, on page 212*.

Configuring Traffic Edge to Use a SOCKS Firewall

You can configure Traffic Edge to use a SOCKS firewall either by using Traffic Manager or by editing configuration files manually. Both procedures are provided below.

▼ To configure Traffic Edge to use a SOCKS firewall from Traffic Manager:

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. Click the **Configure** tab.

   The **General** tab of the **My Proxy**/**Basic** button displays.
3. In the **Features** table on the **General** tab, click the **SOCKS On** button in the **Security** section.
4. Click the **Apply** button.
5. On the **Configure** tab, click the **Security** button and then click the **SOCKS** button.
6. In the **SOCKS Version** area on the **General** tab, specify the SOCKS version running on your SOCKS servers.
7. Click the **Apply** button.
8. Click the **Server** tab.
9. In the **Default Servers** field of the **SOCKS Server** section, enter the hostnames of your default SOCKS servers and the ports through which Traffic Edge communicates with the SOCKS servers. Separate the hostname and the port with a colon (:) and separate each entry with a semicolon (;): for example, `socks1:1080;socks2:4080`.
10. Click the **Apply** button.
11. In the **SOCKS Server Rules** area, click the **Edit File** button to perform additional SOCKS server configuration, such as SOCKS server bypass and authentication.

    The configuration file editor for the `socks.config` file opens.
12. Enter information in the fields provided and then click the **Add** button. All the fields are described in *Appendix B, Traffic Manager Configuration Options*.
13. Click the **Apply** button to save the information and then click the **Close** button to exit the configuration file editor.
14. Restart Traffic Edge by clicking the **Restart** button on the **Configure**/**My Proxy**/**Basic/General** tab.

▼ **To configure Traffic Edge to use a SOCKS firewall manually:**

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Edit the following variables:

Variable	Description
proxy.config.socks.socks_needed	Set this variable to 1 to enable SOCKS.
proxy.config.socks.socks_version	Set this variable to the SOCKS version supported by your SOCKS servers. Traffic Edge supports SOCKS Version 4 and Version 5.
proxy.config.socks.socks.default_servers	Specify the hostnames of your default SOCKS servers and the ports through which Traffic Edge communicates with the SOCKS servers: for example, `socks1:1080;socks2:4080` You can perform additional SOCKS server configuration in the `socks.config` file. You can specify that requests to specific origin servers go through specific SOCKS servers; refer to *socks.config, on page 451*.

3. Save and close the `records.config` file.
4. In a text editor, open the `socks.config` file located in the Traffic Edge `config` directory.
5. Enter a line in the file specifying the IP addresses or IP address range of the origin servers that you want Traffic Edge to access directly; refer to *socks.config, on page 451*.
6. Save and close the `socks.config` file.
7. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
8. Run the command `traffic_line -L` to restart Traffic Edge on the local node or `traffic_line -M` to restart Traffic Edge on all the nodes in a cluster.

Setting SOCKS Proxy Options

To configure Traffic Edge as a SOCKS proxy, you must enable the SOCKS proxy option and specify the port on which Traffic Edge accepts SOCKS traffic from SOCKS clients.

As a SOCKS proxy, Traffic Edge can receive SOCKS packets (usually on port 1080) from the client. By inspecting the packets, Traffic Edge detects and serves HTTP requests but forwards all other requests directly to the SOCKS server.

Note You must set SOCKS proxy options in addition to enabling the SOCKS option and specifying SOCKS server information described in *Configuring Traffic Edge to Use a SOCKS Firewall, on page 211*.

You can set SOCKS proxy options either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ **To set SOCKS proxy options from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Security** button and then click the **SOCKS** button.
3. Click the **Proxy** tab.
4. Enable the **SOCKS Proxy** option.
5. In the **SOCKS Proxy Port** field, specify the port on which Traffic Edge accepts SOCKS traffic. The default is port 1080.
6. Click the **Apply** button.
7. Restart Traffic Edge by clicking the **Restart** button on the **Configure**/**My Proxy**/**Basic/General** tab.

▼ **To set SOCKS proxy options manually:**

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Edit the following variables:

Variable	Description
proxy.config.socks.accept_enabled	Set this variable to 1 to enable the SOCKS proxy option. Change this variable only if you want to use Traffic Edge as a SOCKS proxy.
proxy.config.socks.accept_port	Specify the port on which Traffic Edge accepts SOCKS traffic. This is usually port 1080.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -L` to restart Traffic Edge on the local node or `traffic_line -M` to restart Traffic Edge on all the nodes in a cluster.

Configuring DNS Server Selection (Split DNS)

The Split DNS option lets you configure Traffic Edge to use multiple DNS servers, depending on your security requirements; for example, you can configure Traffic Edge to look to one set of DNS servers to resolve hostnames on your internal network, while allowing DNS servers outside the firewall to resolve hosts on the Internet. This maintains the security of your intranet, while continuing to provide direct access to sites outside your organization.

To configure Split DNS, you must perform the following tasks:

- Specify the rules for performing DNS server selection based on the destination domain, the destination host, or a URL regular expression.
- Enable the Split DNS option.

You can configure Split DNS by either using Traffic Manager or by editing configuration files manually. Both procedures are provided below.

▼ To configure Split DNS from Traffic Manager:

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Networking** button and then click the **DNS Resolver** button.
3. Click the **Split DNS** tab.
4. Enable the **Split DNS** option.
5. In the **Default Domain** field, enter the default domain for split DNS requests. Traffic Edge appends this value automatically to a hostname that does not include a domain before determining which DNS server to use.
6. In the **DNS Servers Specification** area, click the **Edit File** button.

   The configuration file editor for the `splitdns.config` file opens.
7. Enter information in the fields provided and then click the **Add** button. All the fields are described in *Appendix B, Traffic Manager Configuration Options*.
8. Click the **Apply** button to save the information and then click the **Close** button to exit the configuration file editor.
9. On the **Split DNS** tab, click the **Apply** button to save your configuration.

▼ To configure Split DNS manually:

1. In a text editor, open the `splitdns.config` file located in the Traffic Edge `config` directory.
2. Add rules to the `splitdns.config` file. For information about the format of the `splitdns.config` file, refer to *page 452*.
3. Save and close the `splitdns.config` file.
4. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
5. Edit the following variables:

Variable	Description
proxy.process.dns.splitDNS.enabled	Set this variable to 1 to enable split DNS.
proxy.config.dns.splitdns.def_domain	Set this variable to specify the default domain for split DNS requests. Traffic Edge appends this value automatically to a hostname that does not include a domain before determining which DNS server to use.

6. Save and close the `records.config` file.
7. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
8. Run the command `traffic_line -x` to apply the configuration changes.

Configuring Proxy Authentication

Traffic Edge supports LDAP, RADIUS, and NTLM proxy authentication.

Traffic Edge supports proxy authentication in both forward and reverse proxy mode. If you have configured Traffic Edge to run in both reverse proxy *and* forward proxy mode, users might be prompted for a username and password more than once to retrieve a document.

IMPORTANT Traffic Edge supports HTTP proxy authentication for Real Networks only with the RealOne player.

Using LDAP Proxy Authentication

Traffic Edge provides the LDAP option to ensure that users are authenticated with an LDAP server before accessing content through Traffic Edge.

When you enable the LDAP option, Traffic Edge acts as an LDAP client and directly challenges users that request content for a username and password. After receiving the username and password from a client, Traffic Edge contacts the LDAP server to check that they are correct. If the LDAP server *accepts* the username and password, Traffic Edge serves the client with the requested content and stores the username and password entry in the Traffic Edge authentication cache; all future authentication requests for that user are served from the authentication cache until the entry expires (after 3600 seconds by default). If the LDAP server *rejects* the username and password, the browser displays a message indicating that authorization failed and prompts again for a username and password.

To configure Traffic Edge to be an LDAP client, you perform the following tasks:

- Enable the LDAP option and specify the hostname of the LDAP server, the port that Traffic Edge uses to communicate with the LDAP server, and the base Distinguished Name (DN); refer to *Configuring Traffic Edge to Be an LDAP Client*, below.
- Set authentication rules to determine which users must be authenticated to access particular sites on the Internet; refer to *Setting LDAP Authentication Rules, on page 217*.

IMPORTANT For Real Networks, you must perform an additional configuration procedure to enable proxy authentication; refer to *Enabling Proxy Authentication for Real Networks, on page 225*.

WMT users are prompted for a username and password more than once when requesting ASX files. Inktomi recommends that WMT users check the **Remember this username and password** option in the first authentication dialog box to prevent further checks.

Configuring Traffic Edge to Be an LDAP Client

You can configure LDAP proxy authentication either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ **To configure LDAP authentication from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. Click the **Configure** tab.

   The **General** tab of the **My Proxy**/**Basic** button displays.

3 In the **Features** table, click **LDAP On** in the **Security** section.

4 Click the **Apply** button.

5 On the **Configure** tab, click the **Security** button and then click the **Access Control** button.

6 Click the **LDAP** tab.

7 Enable the **Purge Cache on Authentication Failure** option to configure Traffic Edge to delete the authorization entry for the client in the Traffic Edge authentication cache if authorization fails.

8 In the **Hostname** field of the **LDAP Server** area, enter the hostname of the LDAP server.

9 In the **Port** field, enter the port on which Traffic Edge communicates with the LDAP server. The default is port 389.

10 In the **Base Distinguished Name** field, enter the base Distinguished Name (DN). Obtain this value from your LDAP administrator.

11 Click the **Apply** button.

12 Restart Traffic Edge by clicking the **Restart** button on the **Configure**/**My Proxy**/**Basic/General** tab.

▼ To configure LDAP authentication manually:

1 In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.

2 Edit the following variables:

Variable	Description
proxy.config.ldap.auth.enabled	Set this variable to 1 to enable LDAP proxy authentication.
proxy.config.ldap.proc.ldap.server.name	Set this variable to specify the hostname of the LDAP server.
proxy.config.ldap.proc.ldap.server.port	Set this variable to specify the LDAP port number. The default port number is 389.
proxy.config.ldap.proc.ldap.base.dn	Set this variable to specify the name of the base Distinguished Name (DN). Obtain this value from your LDAP administrator. You must specify a correct base DN; otherwise, LDAP authentication will fail to operate.
proxy.config.ldap.auth.purge_cache_on_auth_fail	Set this variable to 1 to configure Traffic Edge to delete the authorization entry for the client in the Traffic Edge authentication cache if authorization fails.

3 Save and close the `records.config` file.

4 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

5 Run the command `traffic_line -L` to restart Traffic Edge on the local node or `traffic_line -M` to restart Traffic Edge on all the nodes in a cluster.

Setting LDAP Authentication Rules

You can set LDAP authentication rules to determine which users must be authenticated to access particular sites on the Internet. You set LDAP authentication rules either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ To set LDAP authentication rules from Traffic Manager:

1 From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.

2 On the **Configure** tab, click the **Security** button and then click the **Access Control** button.

3 In the **Filtering** area on the **Filtering** tab, click the **Edit File** button.

The configuration file editor for the `filter.config` file opens.

4 From the **Rule Type** drop-down list box, select **ldap**.

5 From the **Primary Destination Type** drop-down list box, select:

- **dest_domain** to base the rule on the origin server domain
- **dest_host** to base the rule on the origin server hostname
- **dest_ip** to base the rule on the origin server IP address
- **url_regex** to base the rule on a regular expression

6 In the **Primary Destination Value** field, enter the primary destination value to which the rule applies; for example, if you selected **dest_ip** from the **Primary Destination Type** drop-down list box, enter the IP address of the origin server.

Optional 7 In the **Secondary Specifiers** area, enter information in the fields provided. All the fields are described in *filter.config Configuration File Editor, on page 328*.

Optional 8 In the **Authentication and Authorization** area, enter information in any of the fields provided. All the fields are described in *filter.config Configuration File Editor, on page 328*.

If you do not specify a server name, Traffic Edge uses the default server specified in **LDAP Server** field of the **LDAP** tab under **Configure/Security/Access Control**.

If you specify any one of the **Authentication and Authorization Specifiers**, you must also specify the **Server Name**, **Base Distinguished Name**, and **UID Filter**.

IMPORTANT For WMT (HTTP streaming), do not enter a value for the **Realm** field that contains the string `server`: otherwise, proxy authentication with the Windows Media Server does not work correctly.

9 Click the **Apply** button to save the information and then click the **Close** button to exit the configuration file editor.

▼ **To set LDAP authentication rules manually:**

1. In a text editor, open the `filter.config` file located in the Traffic Edge `config` directory.
2. Add rules to the `filter.config` file. For information about the format of the `filter.config` file, refer to *page 452*.
3. Save and close the `filter.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -L` to restart Traffic Edge on the local node or `traffic_line -M` to restart Traffic Edge on all the nodes in a cluster.

Using RADIUS Proxy Authentication

Traffic Edge provides the RADIUS option to ensure that users are authenticated with a RADIUS server before accessing content through Traffic Edge.

When you enable the RADIUS option, Traffic Edge acts as a RADIUS client and directly challenges users that request content for a username and password. After receiving the username and password from a client, Traffic Edge contacts the RADIUS server to check that they are correct. If the RADIUS server *accepts* the username and password, Traffic Edge serves the client with the requested content and stores the username and password entry in the Traffic Edge authentication cache; all future authentication requests for that user are served from the authentication cache until the entry expires. If the RADIUS server *rejects* the username and password, the browser displays a message indicating that authorization failed and prompts again for a username and password.

Traffic Edge supports a primary RADIUS server and a secondary RADIUS server for failover. If the Primary server does not respond to the Traffic Edge request within the specified timeout (60 seconds by default) Traffic Edge tries to check the username and password again. If a response from the primary RADIUS server is not received after the maximum number of retries (10 by default), Traffic Edge contacts the Secondary RADIUS server. If Traffic Edge cannot contact the secondary RADIUS server, the user is prompted again for a username and password.

The Traffic Edge authentication cache is held in memory and stored on disk. In addition, Traffic Edge stores username and password entries in the authentication cache for 60 minutes. If a password and username entry is expired in the authentication cache, Traffic Edge contacts the RADIUS server to accept or reject the username and password.

To configure Traffic Edge to be a RADIUS client, you perform the following tasks:

- Enable the RADIUS option and specify the hostname or IP address of the primary and secondary (optional) RADIUS servers, as well as the port and shared key that Traffic Edge uses to communicate with the RADIUS servers; refer to *Configuring Traffic Edge to Be a RADIUS Client, on page 219*.
- Set authentication rules to determine which users must be authenticated to access particular sites on the Internet; refer to *Setting RADIUS Authentication Rules, on page 220*.

Configuring Traffic Edge to Be a RADIUS Client

You can configure Traffic Edge to be a RADIUS client either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

In addition to performing the procedure described below, you must add the Traffic Edge machine as a trusted client on the primary and secondary RADIUS servers and provide the shared key you want to use for the Traffic Edge machine (the shared key must be the same one you specify in the following procedure). Refer to your RADIUS server documentation.

▼ **To configure RADIUS authentication from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. Click the **Configure** tab.

   The **General** tab of the **My Proxy**/**Basic** button displays.
3. In the **Features** table, click the **Radius On** button in the **Security** section.
4. Click the **Apply** button.
5. On the **Configure** tab, click the **Security** button and then click the **Access Control** button.
6. Click the **Radius** tab.
7. In the **Hostname** field of the **Primary Radius Server** area, enter the hostname of your primary RADIUS server.
8. In the **Port** field, enter the port number through which Traffic Edge communicates with the primary RADIUS server.
9. In the **Shared Key** field, enter the key used for encoding.
10. If you are using a secondary RADIUS server, enter the hostname, port, and shared key in the appropriate fields of the **Secondary Radius Server (Optional)** area.
11. Click the **Apply** button.
12. Restart Traffic Edge by clicking the **Restart** button on the **Configure**/**My Proxy**/**Basic/General** tab.

▼ **To configure RADIUS authentication manually:**

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Edit the following variables:

Variable	Description
proxy.config.radius.auth.enabled	Set this variable to 1 to enable RADIUS proxy authentication.
proxy.config.radius.proc.radius.primary_server.name	Set this variable to specify the hostname of your primary RADIUS server.

Variable	Description
proxy.config.radius.proc.radius.primary_server.auth_port	Set this variable to specify the port number through which Traffic Edge communicates with the primary RADIUS server.
proxy.config.radius.proc.radius.primary_server.shared_key	Set this variable to specify the key used for encoding.
proxy.config.radius.proc.radius.secondary_server.name	Set this variable to specify the hostname of your secondary RADIUS server.
proxy.config.radius.proc.radius.secondary_server.auth_port	Set this variable to specify the port number through which Traffic Edge communicates with the secondary RADIUS server.
proxy.config.radius.proc.radius.secondary_server.shared_key	Set this variable to specify the key used for encoding.

3 Save and close the `records.config` file.

4 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

5 Run the command `traffic_line -L` to restart Traffic Edge on the local node or `traffic_line -M` to restart Traffic Edge on all the nodes in a cluster.

Setting RADIUS Authentication Rules

You can set RADIUS authentication rules to determine which users must be authenticated to access particular sites on the Internet. You set RADIUS authentication rules either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ To set RADIUS authentication rules from Traffic Manager:

1 From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.

2 On the **Configure** tab, click the **Security** button and then click the **Access Control** button.

3 In the **Filtering** area on the **Filtering** tab, click the **Edit File** button.

The configuration file editor for the `filter.config` file opens.

4 From the **Rule Type** drop-down list box, select **radius**.

5 From the **Primary Destination Type** drop-down list box, select:

- **dest_domain** to base the rule on the origin server domain
- **dest_host** to base the rule on the origin server hostname
- **dest_ip** to base the rule on the origin server IP address
- **url_regex** to base the rule on a regular expression

6 In the **Primary Destination Value** field, enter the primary destination value to which the rule applies; for example, if you selected **dest_ip** from the **Primary Destination Type** drop-down list box, enter the IP address of the origin server.

Optional 7 In the **Secondary Specifiers** area, enter information in the fields provided. All the fields are described in *filter.config Configuration File Editor, on page 328*.

Optional 8 In the **Realm** field of the **Authentication and Authorization** area, enter the realm. The default realm is Traffic Edge.

IMPORTANT For WMT (HTTP streaming), do not enter a value that contains the string `server`: otherwise, proxy authentication with the Windows Media Server does not work correctly.

Optional 9 In the **Redirect URL** field of the **Authentication and Authorization** area, enter the URL that Traffic Edge redirects to when an error occurs.

Traffic Edge does not use any of the other fields in the **Authentication and Authorization** area for RADIUS.

10 Click the **Apply** button to save the information and then click the **Close** button to exit the configuration file editor.

▼ **To set RADIUS authentication rules manually:**

1 In a text editor, open the `filter.config` file located in the Traffic Edge `config` directory.

2 Add rules to the `filter.config` file. For information about the format of the `filter.config` file, refer to *page 452*.

3 Save and close the `filter.config` file.

4 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

5 Run the command `traffic_line -L` to restart Traffic Edge on the local node or `traffic_line -M` to restart Traffic Edge on all the nodes in a cluster.

Using NTLM Proxy Authentication

Traffic Edge provides the NTLM (NT LAN Manager) option to ensure that users in a Windows network are authenticated before they access protected content on the Internet.

When you enable the NTLM option, Traffic Edge can challenge users that request content for their credentials. Traffic Edge then sends the credentials directly to the Windows domain controller to be validated. If the credentials are valid, Traffic Edge serves the requested content and stores the credentials in the Traffic Edge authentication cache for future use. If the credentials are not valid, Traffic Edge sends an *authentication failed* message to the user.

Traffic Edge supports both *Single Sign-On* for Microsoft Internet Explorer (MSIE) and Basic authentication for Netscape and other browsers. Single Sign-On allows users to sign on only once so that they can seamlessly access all authorized network resources. Therefore, if a user has already logged on to the Windows network successfully, the credentials specified during Windows logon are used for authentication and the user is not prompted again for a username and password. With Basic authentication, users *are* prompted for a username and password before they can access the protected content.

Traffic Edge supports the use of backup domain controllers for failover. If the primary domain controller does not respond to the Traffic Edge request, Traffic Edge contacts the next domain controller in the list (the backup domain controller). For the next request, Traffic Edge tries to contact the primary domain controller again and then contacts the backup domain controller if the connection fails. Traffic Edge does this five times before considering the server unavailable. After considering the primary domain controller unavailable, Traffic Edge waits five minutes before trying to contact it again.

IMPORTANT Traffic Edge does not support WINS resolution. Domain controllers must have hostnames that can be resolved by a DNS server.

IMPORTANT NTLM does not work if Extended Security is enabled on the Domain Controller.

Note Traffic Edge supports access to Windows NT domain controllers and to the Windows 2000 Active Directory. However, you do not have to add the Traffic Edge machine to the domain Windows NT controllers or to the Windows 2000 Active Directory.

To configure NTLM proxy authentication, you perform the following tasks:

- Enable the NTLM option and specify information about your domain controllers; refer to *Configuring NTLM Proxy Authentication*, below.
- Set authentication and group authorization rules to determine which users must be authenticated and authorized to access particular sites on the Internet; refer to *Setting NTLM Authentication and Authorization Rules, on page 223*.

IMPORTANT For Real Networks, you must perform an additional configuration procedure to enable proxy authentication; refer to *Enabling Proxy Authentication for Real Networks, on page 225*.

Configuring NTLM Proxy Authentication

You can configure NTLM proxy authentication either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ To configure NTLM authentication from Traffic Manager:

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. Click the **Configure** tab.

   The **General** tab of the **My Proxy**/**Basic** button displays.
3. In the **Features** table, click the **NTLM On** button in the **Security** section.
4. Click the **Apply** button.
5. On the **Configure** tab, click the **Security** button and then click the **Access Control** button.
6. Click the **NTLM** tab.
7. In the **Domain Controller Hostnames** field, enter the hostnames of your domain controllers. Separate each entry with a comma.

8 In the **NT Domain Name** field, enter the domain name of the domain controller you want Traffic Edge to authenticate against.

9 Enable **Load Balancing** if you want Traffic Edge to balance the load when sending authentication requests to the domain controllers.

10 Click the **Apply** button.

11 Restart Traffic Edge by clicking the **Restart** button on the **Configure**/**My Proxy**/**Basic/General** tab.

▼ **To configure NTLM authentication manually:**

1 In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.

2 Edit the following variables:

Variable	Description
proxy.config.ntlm.auth.enabled	Set this variable to 1 to enable NTLM proxy authentication.
proxy.config.ntlm.dc.list	Set this variable to specify the hostnames of your Domain Controllers. Separate each entry with a comma.
proxy.config.ntlm.dc.load_balance	Set this variable to 1 to enable Traffic Edge to balance the load when sending authentication requests to the domain controllers.
proxy.config.ntlm.nt_domain	Set this variable to specify the NT domain name you want Traffic Edge to authenticate against.

3 Save and close the `records.config` file.

4 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

5 Run the command `traffic_line -L` to restart Traffic Edge on the local node or `traffic_line -M` to restart Traffic Edge on all the nodes in a cluster.

Setting NTLM Authentication and Authorization Rules

You can set NTLM authentication and group authorization rules to determine which users must be authenticated and authorized to access particular sites on the Internet. You set NTLM authentication and group authorization rules either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

Note For NTLM group authorization, only the Active Directory is supported.

▼ **To set NTLM authentication and group authorization rules from Traffic Manager:**

1 From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.

2 On the **Configure** tab, click the **Security** button and then click the **Access Control** button.

3 In the **Filtering** area on the **Filtering** tab, click the **Edit File** button.

The configuration file editor for the `filter.config` file opens.

4 From the **Rule Type** drop-down list box, select **ntlm**.

5 From the **Primary Destination Type** drop-down list box, select:

- **dest_domain** to base the rule on the origin server domain
- **dest_host** to base the rule on the origin server hostname
- **dest_ip** to base the rule on the origin server IP address
- **url_regex** to base the rule on a regular expression

6 In the **Primary Destination Value** field, enter the primary destination value to which the rule applies; for example, if you selected **dest_ip** from the **Primary Destination Type** drop-down list box, enter the IP address of the origin server.

Optional

7 In the **Secondary Specifiers** area, enter information in any of the fields provided. All the fields are described in *filter.config Configuration File Editor, on page 328*.

8 For group authorization, you must provide the server name, base Distinguished Name, UID filter, attribute name, attribute value, bind DN, and bind password in the fields provided in the **Authentication and Authorization Specifiers** section. The **Realm** and the **URL Redirect** fields are optional.

9 For authentication, you can specify the realm and the redirect URL in the **Authentication and Authorization Specifiers** section. Traffic Edge does not use the remaining fields in the **Authentication and Authorization Specifiers** section for authentication.

IMPORTANT

For WMT (HTTP streaming), do not enter a value for the **Realm** field that contains the string `server`: otherwise, proxy authentication with the Windows Media Server does not work correctly.

10 Click the **Apply** button to save the information and then click the **Close** button to exit the configuration file editor.

▼ To set NTLM authentication and authorization rules manually:

1 In a text editor, open the `filter.config` file located in the Traffic Edge `config` directory.

2 Add rules to the `filter.config` file. For information about the format of the `filter.config` file, refer to *page 452*.

3 Save and close the `filter.config` file.

4 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

5 Run the command `traffic_line -L` to restart Traffic Edge on the local node or `traffic_line -M` to restart Traffic Edge on all the nodes in a cluster.

Enabling Proxy Authentication for Real Networks

For proxy authentication for Real Networks requests, you must use the following procedure in addition to completing the procedures outlined in *Using LDAP Proxy Authentication, on page 215* and *Using NTLM Proxy Authentication, on page 221*.

▼ **To enable proxy authentication for Real Networks:**

1. In a text editor, open the `rmserver.cfg` file located in the Traffic Edge `mproxy` directory. This file configures RealProxy.
2. Locate the following configuration text:

```
<List Name="AuthenticationRealms">
 <List Name="SecureAdmin">
 <Var Realm="delancey.AdminRealm"/>
  <List Name="BasicAuthenticator">
   <Var PluginID="rn-auth-basic"/>
   <Var DatabaseID="Admin_Basic"/>
  </List>
 </List>
 <List Name="ConnectRealm">
  <Var Realm=".ConnectRealm"/>
   <List Name="BasicAuthenticator">
    <Var PluginID="rn-auth-basic"/>
    <Var DatabaseID="Connect_RN5"/>
   </List>
  </List>
</List>
```

3. Change the `PluginID` for the `ConnectRealm` from `"rn-auth-basic"` to `"ink-auth"` so that it appears as follows:

```
<List Name="AuthenticationRealms">
 <List Name="SecureAdmin">
  <Var Realm="delancey.AdminRealm"/>
   <List Name="BasicAuthenticator">
    <Var PluginID="rn-auth-basic"/>
    <Var DatabaseID="Admin_Basic"/>
   </List>
  </List>
  <List Name="ConnectRealm">
   <Var Realm=".ConnectRealm"/>
    <List Name="BasicAuthenticator">
      <Var PluginID="ink-auth"/>
      <Var DatabaseID="Connect_RN5"/>
    </List>
  </List>
</List>
```

4 Locate the following configuration text:

```
<List Name="ProxyAuthentication">
 <Var Enabled="0"/>
  <List Name="Authority">
   <Var DatabaseID="Connect_RN5"/>
   <Var Realm=".ConnectRealm"/>
   <Var AllowDuplicateIDs="0"/>
  </List>
</List>
```

5 Change the `Enabled` value from 0 to 1 so that it appears as follows:

```
<List Name="ProxyAuthentication">
 <Var Enabled="1"/>
  <List Name="Authority">
   <Var DatabaseID="Connect_RN5"/>
   <Var Realm=".ConnectRealm"/>
   <Var AllowDuplicateIDs="0"/>
  </List>
</List>
```

6 Save and close the `rmserver.cfg` file.

7 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

8 Run the command `traffic_line -L` to restart Traffic Edge on the local node or `traffic_line -M` to restart Traffic Edge on all the nodes in a cluster.

Using SSL Termination

The Traffic Edge SSL termination option enables you to secure connections in reverse proxy mode between a client and a Traffic Edge and/or Traffic Edge and an origin server.

The following sections describe how to enable and configure the SSL termination option.

- To enable and configure SSL termination for client/Traffic Edge connections, follow the procedures in *Client and Traffic Edge Connections, on page 227*.
- To enable and configure SSL termination for Traffic Edge/origin server connections, refer to *Traffic Edge and Origin Server Connections, on page 230*.
- To enable and configure SSL termination for both client/Traffic Edge and Traffic Edge/origin server connections, follow the procedures in both *Client and Traffic Edge Connections*, below, and *Traffic Edge and Origin Server Connections, on page 230*.

If you install an SSL accelerator card on your Traffic Edge system, you must perform additional configuration steps; refer to *Configuring Traffic Edge to Use an SSL Accelerator Card, on page 233*.

Client and Traffic Edge Connections

Figure 35 illustrates communication between a client and Traffic Edge, and between Traffic Edge and an origin server when the SSL termination option is enabled and configured for client/Traffic Edge connections only.

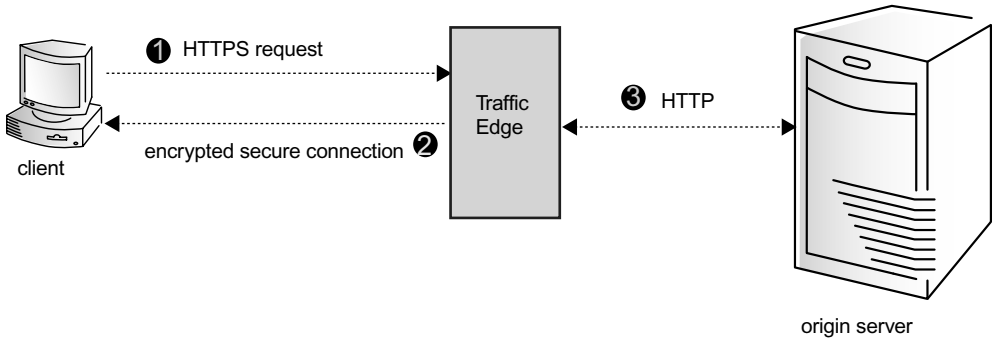


Figure 35 Client and Traffic Edge communication using SSL termination

Figure 35 demonstrates the following steps.

Step 1 The client sends an HTTPS request for content. Traffic Edge receives the request and performs the SSL handshake to authenticate the client (depending on the authentication options configured) and to determine the encryption method to be used. If the client is allowed access, Traffic Edge checks its cache for the requested content.

Step 2 If the request is a cache hit and the content is fresh, Traffic Edge encrypts the content and sends it to the client, where it is decrypted (using the method determined during the handshake) and displayed.

Step 3 If the request is a cache miss or is stale, Traffic Edge communicates with the origin server via HTTP and obtains the plain text version of the content. Traffic Edge saves the plain text version of the content in its cache and then encrypts the content and sends it to the client, where it is decrypted and displayed.

To configure Traffic Edge to use the SSL termination option for client/Traffic Edge connections, you must perform the following procedures:

- Obtain and install an SSL *server* certificate from a recognized certificate authority (such as VeriSign). The SSL server certificate contains information that allows the client to authenticate Traffic Edge and exchange secret encryption keys.
- Configure SSL termination options:
  - Enable the SSL termination option
  - Set the port number used for SSL communication
  - Specify the filename and location of the server certificate

Optional

  - Configure the use of client certificates

    Client certificates are located on the client. If you configure Traffic Edge to require client certificates, Traffic Edge verifies the client certificate during the SSL handshake to authenticate the client. If you configure Traffic Edge to *not* require client certificates, access to Traffic Edge is managed through access control lists and other Traffic Edge options that have been set: for example, rules in the `ip_allow.config` file and LDAP-based proxy authentication.

- Specify the filename and location of the Traffic Edge private key (if the private key is not located in the server certificate file)

  Traffic Edge uses its private key during the SSL handshake to decrypt the session encryption keys. The private key must be stored and protected against theft.

Optional

- Configure the use of certification authorities (CAs)

  CAs provide added security when using client certificates by verifying the identity of the person requesting a certificate.

You can configure the SSL termination option either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ To configure SSL termination for client/Traffic Edge connections from Traffic Manager:

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. Click the **Configure** tab.

   The **General** tab of the **My Proxy**/**Basic** button displays.
3. In the **Features** table, click the **SSL Termination On** button in the **Security** section.
4. Click the **Apply** button.
5. On the **Configure** tab, click the **Security** button and then click the **SSL Termination** button.
6. In the **SSL Termination Port** field on the **General** tab, enter the port Traffic Edge uses for SSL communication.
7. Click the **Apply** button.
8. Click the **Client-Proxy** tab.
9. In the **Client Certificate** section, specify if a client certificate is necessary:
   - Enable **Not Required** if no client certificates are required. Traffic Edge does not verify client certificates during the SSL handshake. Access to Traffic Edge depends on Traffic Edge configuration options (such as access control lists).
   - Enable **Optional** if client certificates are optional. If a client has a certificate, the certificate is validated. If the client does not have a certificate, the client is still allowed access to Traffic Edge unless access is denied through other Traffic Edge configuration options.
   - Enable **Required** if client certificates are required. The client must be authenticated during the SSL handshake. Clients without a certificate are not allowed to access Traffic Edge.
10. In the **Server Certificate File** field, enter the filename of the Traffic Edge SSL server certificate.
11. In the **Server Private Key** field, enter the filename of the Traffic Edge private key.
12. In the **Certificate Authority** field, enter the filename of the certificate authority that client certificates will be verified against.
13. In the **SSL Multi-Certificate** area, edit the `ssl_muticert.config` file if you want to configure Traffic Edge to use multiple SSL server certificates with the SSL termination option. If you have a Traffic Edge system with more than one IP address assigned to it, you can assign a different SSL certificate to be served when a client requests a

particular IP address; refer to *ssl_multicert.config, on page 454* for information about the format of the file.

14 Click the **Apply** button to save your configuration.

15 Restart Traffic Edge by clicking the **Restart** button on the **Configure**/**My Proxy**/**Basic/General** tab.

▼ To configure SSL termination for client/Traffic Edge connections manually:

1 In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.

2 Edit the following variables in the `SSL Termination` section of the file:

Variable	Description
proxy.config.ssl.enabled	Set this variable to 1 to enable the SSL termination option.
proxy.config.ssl.server_port	Set this variable to specify the port used for SSL communication. The default port is 443.
proxy.config.ssl.client.certification_level	Set this variable to one of the following values: 0 specifies that no client certificates are required. Traffic Edge does not verify client certificates during the SSL handshake. Access to Traffic Edge depends on Traffic Edge configuration options (such as access control lists). 1 specifies that client certificates are optional. If a client has a certificate, the certificate is validated. If the client does not have a certificate, the client is still allowed access to Traffic Edge unless access is denied through other Traffic Edge configuration options. 2 specifies that client certificates are required. The client must be authenticated during the SSL handshake. Clients without a certificate are not allowed to access Traffic Edge.
proxy.config.ssl.server.cert.filename	Set this variable to specify the filename of the Traffic Edge SSL server certificate. Traffic Edge provides a demo server certificate called `server.pem`. You can use this certificate to verify that the SSL feature is working. If you are using multiple server certificates, set this variable to specify the default filename.
proxy.config.ssl.server.cert.path	Set this variable to specify the location of the Traffic Edge SSL server certificate. The default directory is the Traffic Edge `config` directory.

Variable	Description
proxy.config.ssl.server.private_key.filename	Set this variable to specify the filename of the Traffic Edge private key. Change this variable only if the private key is not located in the Traffic Edge SSL server certificate file.
proxy.config.ssl.server.private_key.path	Set this variable to specify the location of the Traffic Edge private key. Change this variable only if the private key is not located in the Traffic Edge SSL server certificate file.
proxy.config.ssl.CA.cert.filename	Specify the filename of the certificate authority that client certificates will be verified against.The default value is `NULL`.
proxy.config.ssl.CA.cert.path	Specify the location of the certificate authority file that client certificates will be verified against.The default value is `NULL`

3 Save and close the `records.config` file.

4 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

5 Run the command `traffic_line -L` to restart Traffic Edge on the local node or `traffic_line -M` to restart Traffic Edge on all the nodes in a cluster.

Traffic Edge and Origin Server Connections

Figure 36 illustrates communication between Traffic Edge and an origin server when the SSL termination option is enabled for Traffic Edge /origin server connections.

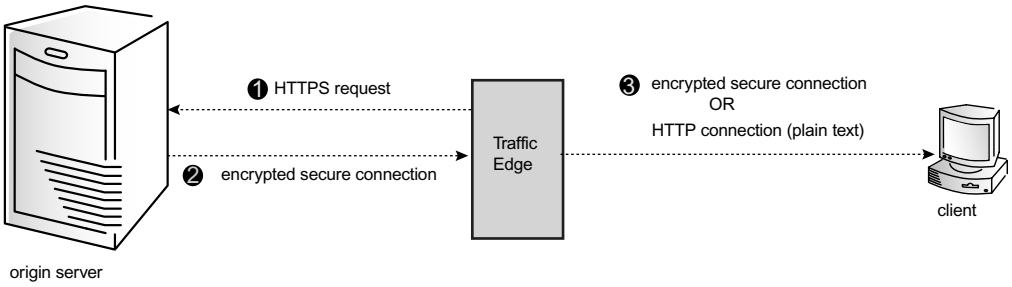


Figure 36 Traffic Edge and origin server communication using SSL termination

Figure 36 demonstrates the following steps.

Step 1 If a client request is a cache miss or is stale, Traffic Edge sends an HTTPS request for the content to the origin server. The origin server receives the request and performs the SSL handshake to authenticate Traffic Edge and to determine the encryption method to be used.

Step 2 If Traffic Edge is allowed access, the origin server encrypts the content and sends it to Traffic Edge, where it is decrypted (using the method determined during the handshake) and the plain text version of the content saved in the cache.

Step 3 If SSL termination is enabled for client /Traffic Edge connections, Traffic Edge re-encrypts the content and sends it to the client via HTTPS, where it is decrypted and displayed. If SSL termination is not enabled for client/Traffic Edge connections, Traffic Edge sends the plain text version of the content to the client via HTTP.

To configure Traffic Edge to use the SSL termination option for Traffic Edge and origin server connections, you must perform the following procedures:

- Obtain and install an SSL *client* certificate from a recognized certificate authority (such as VeriSign). The SSL client certificate contains information that allows the origin server to authenticate Traffic Edge.

  The client certificate is optional.

- Configure SSL termination options:
  - Enable the SSL termination option
  - Set the port number used for SSL communication
  - Specify the filename and location of the SSL client certificate (if you choose to use a client certificate)
  - Specify the filename and location of the Traffic Edge private key (if the private key is not located in the client certificate file)

    Traffic Edge uses its private key during the SSL handshake to decrypt the session encryption keys. The private key must be stored and protected against theft.
  - Configure the use of CAs

    CAs allows the Traffic Edge that is acting as a client to verify the identity of the server with which it is communicating and to exchange secret encryption keys.

You can configure SSL termination options either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ To configure SSL termination for Traffic Edge/origin server connections from Traffic Manager:

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. Click the **Configure** tab.

   The **General** tab of the **My Proxy**/**Basic** button displays.
3. In the **Features** table, click the **SSL Termination On** button in the **Security** section.
4. Click the **Apply** button.
5. In the **SSL Termination Port** field on the **General** tab, enter the port Traffic Edge uses for SSL communication.
6. Click the **Apply** button.
7. Click the **Proxy-Server** tab.
8. Enable the **Certificate Verification** option if you require Traffic Edge to verify the origin server certificate with the CA.

9 In the **Client Certificate File** field, enter the filename of the client certificate (if you have installed an SSL client certificate on Traffic Edge).

10 In the **Client Private Key** field, enter the filename of the Traffic Edge private key.

11 In the **Certificate Authority** field, enter filename of the certificate authority against which the origin server will be verified.

12 Click the **Apply** button.

13 Restart Traffic Edge by clicking the **Restart** button on the **Configure**/**My Proxy**/**Basic/General** tab.

▼ To configure SSL termination for Traffic Edge/origin server connections manually:

1 In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.

2 Edit the following variables in the `SSL Termination` section of the file:

Variable	Description
proxy.config.ssl.auth.enabled	Set this variable to 1 to enable the SSL termination option.
proxy.config.ssl.server_port	Set this variable to specify the port used for SSL communication. The default port is 443.
proxy.config.ssl.client.verify.server	Set this option to 1 to require Traffic Edge to verify the origin server certificate with the CA.
proxy.config.ssl.client.cert.filename	If you have installed an SSL client certificate on Traffic Edge, set this variable to specify the filename of client certificate.
proxy.config.ssl.client.cert.path	If you have installed an SSL client certificate on Traffic Edge, set this variable to specify the location of the client certificate. The default directory is the Traffic Edge `config` directory.
proxy.config.ssl.client.private_key.filename	Set this variable to specify the filename of the Traffic Edge private key. Change this variable only if the private key is not located in the Traffic Edge SSL client certificate file.
proxy.config.ssl.client.private_key.path	Set this variable to specify the location of the Traffic Edge private key. Change this variable only if the private key is not located in the SSL client certificate file.
proxy.config.ssl.client.CA.cert.filename	Specify the filename of the certificate authority against which the origin server will be verified.The default value is `NULL`.
proxy.config.ssl.client.CA.cert.path	Specify the location of the certificate authority file against which the origin server will be verified.The default value is `NULL`.

3 Save and close the `records.config` file.

4 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

5 Run the command `traffic_line -L` to restart Traffic Edge on the local node or `traffic_line -M` to restart Traffic Edge on all the nodes in a cluster.

Configuring Traffic Edge to Use an SSL Accelerator Card

You can install an SSL accelerator card on your Traffic Edge machine to accelerate the number of requests Traffic Edge can process. Traffic Edge supports the following SSL accelerator cards:

- The nCipher nFast card
- The Rainbow CryptoSwift card
- The Compaq Atalla card

After you have installed one of the supported SSL accelerator cards, you must perform the following configuration steps either from Traffic Manager or manually by editing a configuration file.

▼ To configure Traffic Edge to use an SSL accelerator card from Traffic Manager:

1 From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.

2 On the **Configure** tab, click the **Security** button and then click the **SSL Termination** button.

3 Click the **Accelerator** tab.

4 Select the accelerator card installed on your Traffic Edge system.

5 If you did not use the default path when you installed the accelerator card, specify the path for the accelerator files in the **Library Path** field.

6 Click the **Apply** button.

7 Restart Traffic Edge by clicking the **Restart** button on the **Configure**/**My Proxy**/**Basic/General** tab.

▼ **To configure Traffic Edge to use an SSL accelerator card manually:**

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Edit the following variables in the `SSL Termination` section of the file:

Variable	Description
proxy.confg.ssl.accelerator.type	Set this variable to specify the type of SSL accelerator card installed on your Traffic Edge machine: 0 = none (no SSL accelerator card is installed on the Traffic Edge machine. The CPU on the Traffic Edge machine determines the number of requests served per second). 1 = nCipher nFast accelerator card 2 = Rainbow CryptoSwift accelerator card 3 = Compaq Atalla accelerator card
proxy.config.ssl.atalla.lib.path	Set this variable to specify the library path for the Compaq Atalla accelerator card. You need only change this variable if you did not use the default path when you installed the card.
proxy.config.ssl.ncipher.lib.path	Set this variable to specify the library path for the nCipher nFast accelerator card. You need only change this variable if you did not use the default path when you installed the card.
proxy.config.ssl.cswift.lib.path	Set this variable to specify the library path for the Rainbow CryptoSwift accelerator card. You need only change this variable if you did not use the default path when you installed the card.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -L` to restart Traffic Edge on the local node or `traffic_line -M` to restart Traffic Edge on all the nodes in a cluster.

Firewall Support for Streaming Media

To allow QuickTime Players and Windows Media Players to play streams, the firewall must allow packets that are bound for certain ports to pass to the inner network. Typically, firewalls are one of the following types:

- An *application-level firewall* that acts as a proxy between the client and the origin server. Because both are proxies, Traffic Edge and the application-level firewall must be configured as a hierarchical proxy, where Traffic Edge is the child and the firewall is the parent. Application-level firewalls must also understand the protocol for the streaming media: HTTP, MMS, RTSP.
- *A network-level firewall* where the firewall must allow packets that are bound for the ports associated with the relevant protocols to pass to the outer network. The main streaming media protocol and port associations are: port 554 for RTSP and port 1755 for MMS.

Traffic Edge uses TCP as the underlying transport protocol for communication with an origin QuickTime or WMT server for splitting live streams. Even if the firewall does not permit in-bound UDP packets, it does not affect streaming for QuickTime or WMT.

Traffic Edge supports the SOCKS option for QuickTime and WMT streaming media, but not for RealProxy. For information about configuring Traffic Edge to use a SOCKS firewall, refer to *Configuring SOCKS Firewall Integration, on page 210*.

Configuring Network-Level Firewalls for Real Networks

When you configure a network-level firewall for use with Traffic Edge, the ports used by your deployment's streaming protocols are open.

To allow the media player to play through the firewall, configure your router to allow packets that are bound for the following ports to pass to the inner network:

- TCP port 7070 for PNA connections to G2 RealServers or pre-G2 RealServers
- TCP port 554 for RTSP connections to G2 RealServers
- TCP port 7802 for the Traffic Edge connection to the Inktomi caching plugin on the origin RealServer
- TCP port 3030 for RealProxy live splitting
- UDP ports 6970 through 7170, inclusive, for incoming traffic only
- TCP port 7802

To set other firewall configuration options, choose one of the following options according to the design parameters of your network:

- Open ports 6970 - 7170 in your firewall for UDP.

  Open ports 7070 - 7071 and 554 in your firewall for TCP. Players will try UDP first, then try TCP afterwards (this option degrades playback quality).

  If your clients use RealPlayer Version 4.0 or 5.0 exclusively, it is not necessary to open port 554. If you have even one client using RealPlayer G2, that client can not function properly unless port 554 of the firewall is open.

- Configure your firewall to receive UDP through only one port and instruct players to use UDP with the port you chose.
- Instruct users to configure RealPlayer to request that RealServer send all media in HTTP format.

  This creates more overhead on your network than any of the other options.

To make your firewall even more secure, configure the firewall's access control list to allow TCP connections on port 7070 and/or port 554 to be initiated from the inside network exclusively. Allow incoming traffic only if it is part of an ongoing connection.

Using the Inktomi Antivirus Extension

The Inktomi Antivirus Extension works together with Symantec CarrierScan Server to prevent viruses from entering your network. When Traffic Edge receives a request for an HTTP object that is not in the cache, it retrieves the object from the origin server. Based on the virus-scanning policy specified, Traffic Edge sends the object to CarrierScan Server to be scanned, and repaired if necessary, before sending the response to the client. Traffic Edge also caches a virus-free copy to satisfy future requests.

Traffic Edge with the Antivirus Extension is typically deployed at each Internet access point on your network and can be configured to work with one or a cluster of CarrierScan Servers.

The Antivirus Extension uses the automatic load-balancing feature of the CarrierScan API to support communication with a cluster of CarrierScan Servers; refer to the CarrierScan Server documentation for more information.

IMPORTANT If you use Traffic Edge in a cache hierarchy, you can install the Antivirus Extension on both the parent and child caches. If you install the extension on the parent cache only, you must configure the child caches to send all cache misses to the parent. The parent sends all new requested objects to CarrierScan Server for scanning and sends only virus-free objects to the child caches.

IMPORTANT Virus scanning is a resource-intensive operation that affects Traffic Edge performance for both throughput and latency. You must ensure that the machine running Traffic Edge and CarrierScan Server is correctly sized to process the volume of anticipated traffic. You must adjust the machine based on the estimated cache hit rate (a lower cache hit rate requires more resources) and on the estimated percentage of objects that need to be scanned based on antivirus policy configurations (a higher percentage of objects requires more resources).

The Antivirus Extension is part of Traffic Edge Security Edition. To install the extension, refer to the *Traffic Edge Installation Guide*.

Configuring the Antivirus Extension

To configure the Antivirus Extension, use the following procedure.

▼ **To configure the Antivirus Extension:**

1. Stop Traffic Edge; refer to *Stopping Traffic Edge, on page 30*.
2. Navigate to the Traffic Edge `config/plugins` directory.
3. In a text editor, open the `vscan.config` file.
4. Edit the variables in the file. Refer to *vscan.config, on page 458* for a description of the variables.
5. Save and close the `vscan.config` file.
6. In a text editor, open the `extensions.config` file.
7. Edit the default list of extensions. Delete those you do not want and add those you do, each on a separate line and without the preceding dot (.). To scan files with no extension, add the line `no_extension`; refer to *extensions.config, on page 374*.
8. Save and close the `extensions.config` file.
9. In a text editor, open the `trusted-host.config` file.
10. Edit the file to list the destination hosts that are *trusted sources*. Traffic Edge bypasses virus scanning for objects requested from these hosts. Enter each destination on a separate line, without the `http://` prefix. For information about the format of the file, refer to *trusted-host.config, on page 456*.
11. Save and close the `trusted-host.config` file.
12. Restart Traffic Edge; refer to *Starting Traffic Edge, on page 25*.

Viewing the Antivirus Extension Log Files

The Antivirus Extension generates two log files in the Traffic Edge `logs` directory:

- `vscan.log`, which records the scanning results for each object sent to CarrierScan Server
- `vscan_stats.log`, which records statistics such as the total number of scanning transactions, the average size of the scanned objects, and the number of objects that were scanned and found to be clean, infected, and repaired

For more information about the `vscan.log` and `vscan_stats.log` files, refer to *Understanding the Antivirus Extension Log Files, on page 280*.

You can configure CarrierScan Server to log important events and to send SMTP email and SNMP alerts for these events; refer to the CarrierScan Server documentation.

CHAPTER 15 Working with Log Files

Traffic Edge generates log files that contain information about every request it receives and every error it detects.

This chapter discusses the following topics:

Understanding Traffic Edge Log Files

Traffic Edge records information about every transaction (or request) that it processes and every error that it detects in log files. Traffic Edge keeps three types of log files:

- *Error log files* record information about why a particular transaction was in error.
- *Event log files* (also called *access log files*) record information about the state of each transaction that Traffic Edge processes.
- *System log files* record system information, which includes messages about the state of Traffic Edge and any errors or warnings that it produces. This kind of information might include a note that event log files were rolled, a warning that cluster communication timed out, or an error indicating that Traffic Edge was restarted. Traffic Edge posts alarms signifying error conditions in Traffic Manager; refer to *Working with Traffic Manager Alarms, on page 186*, for details.

  In UNIX, all system information messages are logged with the system-wide logging facility `syslog` under the daemon facility. The `syslog.conf` configuration file (stored in the `/etc` directory) specifies where these messages are logged. A typical location is `/var/adm/messages` (Solaris) or `/var/log/messages` (Linux).

  The `syslog` process works on a system-wide basis, so it serves as the single repository for messages from all Traffic Edge processes, including `traffic_server`, `traffic_manager`, and `traffic_cop`.

  In Windows, system information messages from the `traffic_server` and `traffic_manager` processes are logged in the application log in the Windows Event Log. To view the application log, select **Administrative Tools**/**Event Viewer** from the **Control Panel** and then click **Application Log** in the tree view. Messages from the `traffic_cop` process are logged to the `cop.log` file (located in the Traffic Edge installation directory).

  System information logs observe a static format. Each log entry in the log contains information about the date and time the error was logged, the hostname of the Traffic Edge that reported the error, and a description of the error or warning.

  Refer to *Appendix F, Traffic Edge Error Messages*, for a list of the system information messages that Traffic Edge logs.

By default, Traffic Edge creates both error and event log files and records system information in system log files. You can disable event logging and/or error logging. From Traffic Manager, click **Configure**/**Subsystems**/**Logging** and select one of the following options: **Log Transactions and Errors**, **Log Transactions Only**, **Log Errors Only**, or **Disabled**. Alternatively, you can set the configuration variable `proxy.config.log2.logging_enabled` in the `records.config` file to one of the following values: `0` to disable both event and error logging, `1` to enable error logging only, `2` to enable transaction logging only, or `3` to enable both transaction and error logging.

Understanding Event Log Files

Event log files record information about every request that Traffic Edge processes. By analyzing the log files, you can determine how many people use the Traffic Edge cache, how much information each person requested, what pages are most popular, and so on.

Traffic Edge supports several standard log file formats, such as Squid and Netscape, and user-defined custom formats. You can analyze the standard format log files with off-the-shelf analysis packages. To help with log file analysis, you can separate log files so that they contain information specific to protocol or hosts. You can also configure Traffic Edge to roll log files automatically at specific intervals during the day or when they reach a certain size.

The following sections describe the Traffic Edge logging system features and discuss how to:

- Manage your event log files

  You can choose a central location for storing log files, set how much disk space to use for log files, and set how and when to roll log files; refer to *Managing Event Log Files, on page 242*.

- Choose different event log file formats

  You can choose which standard log file formats you want to use for traffic analysis: for example, Squid or Netscape. Alternatively, you can use the Traffic Edge custom format, which is XML-based and enables you to institute more control over the type of information recorded in log files. Refer to *Choosing Event Log File Formats, on page 244*.

  Traffic Edge uses the custom format to create event logs that detail streaming media transactions; refer to *Working with Streaming Media Log Files, on page 264*.

- Roll event log files automatically

  You can configure Traffic Edge to roll event log files when they reach a certain size or at specific intervals during the day so that you can identify and manipulate log files that are no longer active; refer to *Rolling Event Log Files, on page 252*.

- Separate log files according to protocols and hosts

  You can configure Traffic Edge to create separate log files for different protocols. You can also configure Traffic Edge to generate separate log files for requests served by different hosts; refer to *Splitting Event Log Files, on page 255*.

- Collate log files from different Traffic Edge nodes

  You can designate one or more nodes on the network to serve as log collation servers. These servers, which might either be standalone or part of Traffic Edge, enable you to keep all logged information in well-defined locations; refer to *Collating Event Log Files, on page 257*.

- View statistics about the logging system

  Traffic Edge provides statistics about the logging system. You can access the statistics through Traffic Manager or through Traffic Line; refer to *Viewing Logging Statistics, on page 267*.

- Interpret log file entries for the log file formats; refer to *Example Event Log File Entries, on page 269*.

Managing Event Log Files

You can manage your event log files and control where they are located, how much space they can consume, and how low disk space in the logging directory is handled.

Choosing the Logging Directory

By default, Traffic Edge writes all event log files in the `logs` directory, which is located in the directory where you installed Traffic Edge. To use a different directory, refer to *Setting Log File Management Options, on page 242*.

Controlling Logging Space

Traffic Edge allows you to control the amount of disk space that the logging directory can consume. This allows the system to operate smoothly within a specified space window for a long period of time.

After you establish a space limit, Traffic Edge continues to monitor the space in the logging directory. When the free space dwindles to the headroom limit (refer to *Setting Log File Management Options, on page 242*), it enters a low space state and takes the following actions:

- If the autodelete option (discussed in *Rolling Event Log Files, on page 252*) is *enabled*, Traffic Edge identifies previously rolled log files (log files with a `.old` extension) and starts deleting files one by one, beginning with the oldest file, until it emerges from the low state. Traffic Edge logs a record of all files it deletes in the system error log.
- If the autodelete option is *disabled* or there are not enough old log files to delete for the system to emerge from its low space state, Traffic Edge issues a warning and continues logging until space is exhausted, at which point it stops event logging. Traffic Edge resumes event logging when enough space becomes available for it to exit its low space state. You can make space available either by removing files from the logging directory manually or by explicitly increasing the logging space limit.

You can run a `cron` script in conjunction with Traffic Edge to automatically remove old log files from the logging directory (before Traffic Edge enters the low space state) and relocate them to a temporary partition. Once the files are relocated, you can run a variety of log analysis scripts on them, after which you can compress the logs and move them to an archive location or delete them.

Setting Log File Management Options

You can set log management options either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ To set log management options from Traffic Manager:

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Subsystems** button and then click the **Logging** button.
3. In the **Log Directory** field, enter the path to the directory in which you want to store event log files. This can be an absolute path or a path relative to the directory in which

Traffic Edge is installed. The default directory is `logs`, located in the Traffic Edge installation directory.

Note The directory you specify must already exist.

4 In the **Limit** field of the **Log Space** area, enter the maximum amount of space you want to allocate to the logging directory. The default value is 2000 MB.

Note All files in the logging directory contribute to the space used, even if they are not log files.

5 In the **Headroom** field of the **Log Space** area, enter the tolerance for the log space limit. The default value is 10 MB.

If the **Auto-Delete Rolled Files** option is enabled in the **Log Rolling** section, autodeletion is triggered when the amount of free space available in the logging directory is less than the headroom. For information about log file rolling, refer to *Rolling Event Log Files, on page 252*.

6 Click the **Apply** button.

▼ To set log management options manually:

1 In a text editor, open the `records.config` file located in the `config` directory.

2 Edit the following variables:

Variable	Description
proxy.config.log2.logfile_dir	Specify the path to the directory in which you want to store event log files. This can be an absolute path or a path relative to the directory in which Traffic Edge is installed. The default is `logs` located in the Traffic Edge installation directory. Note: The directory you specify must already exist.
proxy.config.log2.max_space_mb_for_logs	Enter the maximum amount of space you want to allocate to the logging directory. The default value is 2000 MB. Note: All files in the logging directory contribute to the space used, even if they are not log files.
proxy.config.log2.max_space_mb_headroom	Enter the tolerance for the log space limit. The default value is 10 MB.

3 Save and close the `records.config` file.

4 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

5 Run the command `traffic_line -x` to apply the configuration changes.

Choosing Event Log File Formats

Traffic Edge supports the following log file formats:

- *Standard formats*, such as Squid or Netscape; refer to *Using Standard Formats*, below.
- The Traffic Edge *custom format*; refer to *Using the Custom Format, on page 246*

In addition to the standard and custom log file format, you can choose whether to save log files in *binary* or *ASCII*; refer to *Choosing Binary or ASCII, on page 250*.

IMPORTANT Event log files consume a large amount of disk space. Creating log entries in multiple formats at the same time can consume disk resources very quickly and adversely impact Traffic Edge performance.

Using Standard Formats

The standard log formats include Squid, Netscape Common, Netscape extended, and Netscape Extended-2.

The standard log file formats can be analyzed with a wide variety of off-the-shelf log-analysis packages. You should use one of the standard event log formats unless you need information that these formats do not provide. Refer to *Using the Custom Format, on page 246*.

By default, Traffic Edge is configured to use the Squid log file format for nonstreaming media transactions. Traffic Edge is configured to use the custom file format for streaming media transactions; refer to *Working with Streaming Media Log Files, on page 264*.

Setting Standard Log File Format Options

You can set standard log file format options either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ **To select a standard event log file format from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Subsystems** button and then click the **Logging** button.
3. Click the **Formats** tab.
4. Enable the format you want to use.
5. Select the log file type (ASCII or binary).
6. In the **Filename** field, enter the name you want to use for your event log files.
7. In the **Header** field, enter a text header that will display at the top of the event log files. Leave this field blank if you do not want to use a text header.
8. Click the **Apply** button.

▼ **To select a standard event log file format manually:**

1. In a text editor, open the `records.config` file located in the `config` directory.
2. To use the Squid format, edit the following variables:

Variable	Description
proxy.config.log2.squid_log_enabled	Set this variable to 1 to enable the Squid log file format.
proxy.config.log2.squid_log_is_ascii	Set this variable to 1 to enable ASCII mode. Set this variable to 0 to enable binary mode.
proxy.config.log2.squid_log_name	Enter the name you want to use for Squid event log files. The default is `squid`.
proxy.config.log2.squid_log_header	Enter the header text you want to display at the top of the Squid log files. Enter `NULL` if you do not want to use a header.

3. To use the Netscape Common format, edit the following variables:

Variable	Description
proxy.config.log2.common_log_enabled	Set this variable to 1 to enable the Netscape Common log file format
proxy.config.log2.common_log_is_ascii	Set this variable to 1 to enable ASCII mode. Set this variable to 0 to enable binary mode.
proxy.config.log2.common_log_name	Enter the name you want to use for Netscape Common event log files. The default is `common`.
proxy.config.log2.common_log_header	Enter the header text you want to display at the top of the Netscape Common log files. Enter `NULL` if you do not want to use a header.

4. To use the Netscape Extended format, edit the following variables:

Variable	Description
proxy.config.log2.extended_log_enabled	Set this variable to 1 to enable the Netscape Extended log file format.
proxy.config.log2.extended_log_is_ascii	Set this variable to 1 to enable ASCII mode. Set this variable to 0 to enable binary mode.
proxy.config.log2.extended_log_name	Enter the name you want to use for Netscape Extended event log files. The default is `extended`.
proxy.config.log2.extended_log_header	Enter the header text you want to display at the top of the Netscape Extended log files. Enter `NULL` if you do not want to use a header.

5 To use the Netscape Extended-2 format, edit the following variables:

Variable	Description
proxy.config.log2.extended2_log_enabled	Set this variable to 1 to enable the Netscape Extended-2 log file format.
proxy.config.log2.extended2_log_is_ascii	Set this variable to 1 to enable ASCII mode. Set this variable to 0 to enable binary mode.
proxy.config.log2.extended2_log_name	Enter the name you want to use for Netscape Extended-2 event log files. The default is `extended2`.
proxy.config.log2.extended2_log_header	Enter the header text you want to display at the top of the Netscape Extended-2 log files. Enter `NULL` if you do not want to use a header.

6 Save and close the `records.config` file.

7 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

8 Run the command `traffic_line -x` to apply the configuration changes.

Using the Custom Format

The Traffic Edge XML-based custom log format is more flexible then the standard log file formats, enabling you to institute much more control over the type of information recorded in your log files. You should create a custom log format if you need data for analysis that is not available in the standard formats. You can decide what information to record for each Traffic Edge transaction and create filters to define which transactions to log.

The heart of the XML-based custom logging feature is an XML-based logging configuration file (`logs_xml.config`) that enables you to create very modular descriptions of logging objects. The `logs_xml.config` file uses three types of objects to create custom log files:

- The `LogFormat` object defines the content of the log file using printf-style format strings.
- The `LogFilter` object defines a filter so that you include or exclude certain information from the log file.
- The `LogObject` object specifies all the information needed to produce a log file:
  - The name of the log file (required).
  - The format to be used (required). This can be a standard format (Squid or Netscape) or a previously defined custom format (a previously defined `LogFormat` object).

- The file mode (ASCII, Binary, or ASCII_PIPE). The default is ASCII.

  The ASCII_PIPE mode writes log entries to a UNIX named pipe (a buffer in memory). Other processes can then read the data using standard I/O functions. The advantage of using this option is that Traffic Edge does not have to write to disk, freeing disk space and bandwidth for other tasks.

  When the buffer is full, Traffic Edge drops log entries and issues an error message indicating how many entries were dropped. Traffic Edge writes only complete log entries to the pipe, therefore, only full records are dropped.

- Any filters you want to use (previously defined `LogFilter` objects).
- The collation servers that are to receive the log files.
- The protocols you want to log (if the protocols tag is used, Traffic Edge will only log transactions from the protocols listed; otherwise, all transactions for all protocols are logged).
- The origin servers you want to log (if the servers tag is used, Traffic Edge will only log transactions for the origin servers listed; otherwise, transactions for all origin servers are logged).
- The header text you want the log files to contain. The header text appears at the beginning of the log file, just before the first record.
- The log file rolling options.

To generate a custom log format, you must specify at least one `LogObject` definition. One log file is produced for each `LogObject` definition. You can create a custom log format by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ To generate a custom log format from Traffic Manager:

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Subsystems** button and then click the **Logging** button.
3. Click the **Custom** tab.
4. Enable the **Custom Logging** option.
5. The **Custom Log File Definitions** area displays the `logs_xml.config` file. Add `LogFormat`, `LogFilter`, and `LogObject` specifications to the configuration file.

   For detailed information about the `logs_xml.config` file and associated object specifications, refer to *logs_xml.config, on page 387*.
6. Click the **Apply** button.

▼ To generate a custom log format manually:

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Edit the following variables:

Variable	Description
proxy.config.log2.custom_logs_enabled	Set this variable to 1 to enable custom logging.
proxy.config.log2.xml_logs_config	Make sure this variable is set to 1 (the default value).

3 Save and close the `records.config` file.

4 Open the `logs_xml.config` file located in the Traffic Edge `config` directory.

5 Add `LogFormat`, `LogFilter`, and `LogObject` specifications to the configuration file. For detailed information about the `logs_xml.config` file and associated object specifications, refer to *logs_xml.config, on page 387*.

6 Save and close the `logs_xml.config` file.

7 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

8 Run the command `traffic_line -x` to apply the configuration changes.

Creating Summary Log Files

Traffic Edge performs several hundred operations per second; therefore, event log files can grow quickly to large sizes. Using SQL-like aggregate operators, you can configure Traffic Edge to create summary log files that summarize a set of log entries over a specified period of time. This can significantly reduce the size of the log files generated.

You generate a summary log file by creating a `LogFormat` object in the XML-based logging configuration file (`logs_xml.config`) using the following SQL-like aggregate operators:

- COUNT
- SUM
- AVERAGE
- FIRST
- LAST

You can apply each of these operators to specific fields, over a specified interval.

▼ **To create a summary log file format:**

1 Access the `logs_xml.config` file either from Traffic Manager or from the Traffic Edge `config` directory:

- On the Traffic Manager **Configure** tab, click the **Subsystems** button and then click the **Logging** button. Click the **Custom** tab to display the `logs_xml.config` file.
- In a text editor, open the `logs_xml.config` file located in the Traffic Edge `config` directory.

2 Define the format of the log file as follows:

```
<LogFormat>
  <Name = "summary"/>
  <Format = "%<operator(field)> : %<operator(field)>"/>
  <Interval = "n"/>
</Format>
```

operator is one of the five aggregate operators (COUNT, SUM, AVERAGE, FIRST, LAST). You can specify more than one operator in the format line.

field is the logging field that you want to aggregate.

n is the interval in seconds between summary log entries.

For more information, refer to *logs_xml.config, on page 387*.

The following example format generates one entry every 10 seconds, with each entry summarizing the timestamp of the last entry of the interval, a count of the number of entries seen within that 10-second interval, and the sum of all bytes sent to the client:

```
<LogFormat>
  <Name = "summary"/>
  <Format = "%<LAST(cqts)> : %<COUNT(*)> : %<SUM(psql)>"/>
  <Interval = "10"/>
</Format>
```

IMPORTANT You cannot create a format specification that contains both aggregate operators and regular fields; for example, the following specification would be invalid:

```
<Format = "%<LAST(cqts)> : %<COUNT(*)> : %<SUM(psql)> : %<cqu>"/>
```

3 Define a `LogObject` that uses this format.

4 Save your changes:

- In Traffic Manager, click the **Apply** button.
- In your text editor, save and close the `logs_xml.config` file and then run the command `traffic_line -x` from the Traffic Edge `bin` directory to apply the configuration changes (in *Windows*, access the `bin` directory from a command prompt window).

Choosing Binary or ASCII

You can configure the Traffic Edge to create event log files in either of the following:

- ASCII

  These files are human readable and can be processed using standard, off-the-shelf log-analysis tools. However, Traffic Edge must perform additional processing to create the files in ASCII, resulting in a slight increase in overhead. Also, ASCII files tend to be larger than the equivalent binary files. ASCII log files have a `.log` filename extension by default.

- Binary

  These files have the advantage of generating lower system overhead, as well as generally occupying less space on the disk, depending on the type of information being logged. You must, however, use a converter application before you can read or analyze these files using standard tools. Binary log files use a `.blog` filename extension by default.

While binary log files typically require less disk space, this is not always the case; for example, the value 0 (zero) requires only one byte to store in ASCII but requires four bytes when stored as a binary integer. On the other hand, if you define a custom format that logs IP addresses, a binary log file would only require four bytes of storage per 32-bit address. However, the same IP address stored in dot notation would require around 15 characters (bytes) in an ASCII log file.

For standard log formats, you select Binary or ASCII on the **Configure/Subsystems/Logging/Formats** tab in Traffic Manager; refer to *Setting Standard Log File Format Options, on page 244*. For the custom log format, you specify ASCII or Binary mode in the `LogObject`; refer to *Using the Custom Format, on page 246*.

For custom log files, in addition to the ASCII and binary options, you can also write log entries to a UNIX named pipe (a buffer in memory). Other processes can then read the data using standard I/O functions. The advantage of using this option is that Traffic Edge does not have to write to disk, freeing disk space and bandwidth for other tasks. In addition, writing to a pipe does not stop when logging space is exhausted because the pipe does not use disk space. Refer to *logs_xml.config, on page 387*, for more information about the ASCII_PIPE option.

Before selecting ASCII versus binary for your log files, consider the type of data that will be logged. Try logging for one day using ASCII and then one day using binary. Assuming that the number of requests is roughly the same for both days, you can calculate a rough metric comparing the two formats.

Using logcat to Convert Binary Logs to ASCII

You must convert a binary log file to ASCII before you can analyze it using standard tools.

▼ **To convert a binary log file to ASCII:**

1. Navigate to the directory that contains the binary log file.
2. Make sure that the `logcat` utility is in your path.
3. Enter the following command:

   `logcat` *`options input_filename...`*

   The following table describes the command-line options.

Option	Description
-o output_file	Specifies where the command output is directed.

-a	Automatically generates the output filename based on the input filename. If the input is from stdin, this option is ignored; for example:
	`logcat -a squid-1.blog squid-2.blog squid-3.blog`
	generates
	`squid-1.log, squid-2.log, squid-3.log`

-S	Attempts to transform the input to Squid format, if possible.
-C	Attempts to transform the input to Netscape Common format, if possible.
-E	Attempts to transform the input to Netscape Extended format, if possible.
-2	Attempt to transform the input to Netscape Extended-2 format, if possible.

Note Use only one of the following options at any given time: -S, -C, -E, or -2.

If no input files are specified, `logcat` reads from the standard input (`stdin`). If you do not specify an output file, `logcat` writes to the standard output (`stdout`); for example, to convert a binary log file to an ASCII file, you can use the `logcat` command with either of the following options:

```
logcat binary_file > ascii_file
logcat -o ascii_file binary_file
```

The binary log file is not modified by this command.

Rolling Event Log Files

Traffic Edge provides automatic log file rolling. This means that at specific intervals during the day or when log files reach a certain size, Traffic Edge closes its current set of log files and opens new log files.

Log file rolling offers the following benefits:

- It defines an interval over which log analysis can be performed.
- It keeps any single log file from becoming too large and assists in keeping the logging system within the specified space limits.
- It provides an easy way to identify files that are no longer being used so that an automated script can clean the logging directory and run log analysis programs.

You should roll log files several times a day. Rolling every six hours is a good guideline to follow.

Rolled Log Filename Format

Traffic Edge provides a consistent name format for rolled log files that allows you to easily identify log files.

When Traffic Edge rolls a log file, it saves and closes the old file and starts a new file. Traffic Edge renames the old file to include the following information:

- The format of the file: for example, `squid.log`.
- The hostname of the Traffic Edge that generated the log file.
- Two timestamps separated by a hyphen (`-`). The first timestamp is a lower bound for the timestamp of the first record in the log file. The lower bound is the time when the new buffer for log records is created. Under low load, the first timestamp in the filename can be different from the timestamp of the first entry. Under normal load, the first timestamp in the filename and the timestamp of the first entry are similar. The second timestamp is an upper bound for the timestamp of the last record in the log file (this is normally the rolling time).
- The suffix `.old`, which makes it easy for automated scripts to find rolled log files.

The timestamps have the following format:

```
%Y%M%D.%Hh%Mm%Ss-%Y%M%D.%Hh%Mm%Ss
```

The following table describes the format:

Code	Definition
`%Y`	The year in four-digit format: for example, 2000.
`%M`	The month in two-digit format, from 01-12: for example, 07.
`%D`	The day in two-digit format, from 01-31: for example, 19.
`%H`	The hour in two-digit format, from 00-23: for example, 21.
`%M`	The minute in two-digit format, from 00-59: for example, 52.
`%S`	The second in two-digit format, from 00-59: for example, 36.

The following is an example of a rolled log filename:

```
squid.log.mymachine.20000912.12h00m00s-20000913.12h00m00s.old
```

The logging system buffers log records before writing them to disk. When a log file is rolled, the log buffer might be partially full. If so, the first entry in the new log file will have a timestamp earlier than the time of rolling. When the new log file is rolled, its first timestamp will be a lower bound for the timestamp of the first entry; for example, suppose logs are rolled every three hours, and the first rolled log file is:

```
squid.log.mymachine.19980912.12h00m00s-19980912.03h00m00s.old
```

If the lower bound for the first entry in the log buffer at 3:00:00 is 2:59:47, the next log file, when rolled, will have the following timestamp:

```
squid.log.mymachine.19980912.02h59m47s-19980912.06h00m00s.old
```

The contents of a log file are always between the two timestamps. Log files do not contain overlapping entries, even if successive timestamps appear to overlap.

Rolling Intervals

Log files are rolled at specific intervals relative to a given hour of the day. Two options control when log files are rolled:

- The offset hour, which is an hour between 0 (midnight) and 23
- The rolling interval

Both the offset hour and the rolling interval determine when log file rolling starts. Rolling occurs every rolling interval *and* at the offset hour; for example, if the rolling interval is six hours and the offset hour is 0 (midnight), then the logs will roll at midnight (00:00), 06:00, 12:00, and 18:00 each day. If the rolling interval is 12 hours and the offset hour is 3, then logs will roll at 03:00 and 15:00 each day.

Setting Log File Rolling Options

You can set log file rolling options either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

Note To configure Traffic Edge to roll log files when they reach a certain size, you must use the manual procedure.

▼ **To set log file rolling options from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Subsystems** button and then click the **Logging** button.
3. On the **General** tab, scroll down to the **Log Rolling** section.
4. Enable the **Log Rolling** option.
5. In the **Offset Hour** field, enter a specific time each day you want log file rolling to take place. Traffic Edge forces the log file to be rolled at the offset hour each day.

   You can enter any hour in the range 0 (midnight) to 23.

6 In the **Interval** field, enter the amount of time Traffic Edge enters data in the log files before rotation takes place.

The minimum value is 300 seconds (five minutes). The maximum value is 86400 seconds (one day).

Note If you start Traffic Edge within a few minutes of the next rolling time, rolling might not occur until the following rolling time.

7 Enable the **Auto-Delete Rolled Files** option to enable autodeletion of rolled log files when available space in the log directory is low.

Autodeletion is triggered when the amount of free space available in the log directory is less than the headroom.

8 Click the **Apply** button.

▼ **To set log file rolling options manually:**

1 In a text editor, open the `records.config` file located in the `config` directory.

2 Edit the following variables:

Variable	Description
proxy.config.log2.rolling_enabled	Set this variable to one of the following values: 1 to enable log file rolling at specific intervals during the day. 2 to enable log file rolling when log files reach a specific size. 3 to enable log file rolling at specific intervals during the day or when log files reach a specific size (whichever occurs first). 4 to enable log file rolling at specific intervals during the day when log files reach a specific size (at a specified time if the file is of the specified size).
proxy.config.log2.rolling_size_mb	Specifies the size that log files must reach before rolling takes place.
proxy.config.log2.rolling_offset_hr	Set this variable to the specific time each day you want log file rolling to take place. Traffic Edge forces the log file to be rolled at the offset hour each day.
proxy.config.log2.rolling_interval_sec	Set this variable to the rolling interval in seconds. The minimum value is 300 seconds (5 minutes). The maximum value is 86400 seconds (one day). Note: If you start Traffic Edge within a few minutes of the next rolling time, rolling might not occur until the following rolling time.
proxy.config.log2.auto_delete_rolled_file	Set this variable to 1 to enable autodeletion of rolled files.

3 Save and close the `records.config` file.

4 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

5 Run the command `traffic_line -x` to apply the configuration changes.

You can fine tune log file rolling settings for a custom log file in the `LogObject` specification in the `logs_xml.config` file. The custom log file uses the rolling settings in its `LogObject`, which override the default settings you specify in Traffic Manager or the `records.config` file described above.

Splitting Event Log Files

By default, Traffic Edge uses standard log formats and generates log files that contain HTTP, FTP, and ICP transactions all together in the same file. However, you can enable log splitting if you prefer to log transactions for different protocols in separate log files.

For the standard log file formats (such as Squid or Netscape), Traffic Edge always records HTTP and FTP transactions in the same log file. You cannot generate separate log files for transactions using these two protocols.

ICP Log Splitting

When ICP log splitting is enabled, Traffic Edge records ICP transactions in a separate log file with a name that contains `icp`; for example, if you enable the Squid format, all ICP transactions are recorded in the `squid-icp.log` file.

When you disable ICP log splitting, Traffic Edge records all ICP transactions in the same log file as HTTP and FTP transactions.

HTTP Host Log Splitting

HTTP host log splitting enables you to record HTTP and FTP transactions for different origin servers in separate log files. When HTTP host log splitting is enabled, Traffic Edge creates a separate log file for each origin server listed in the `log_hosts.config` file (refer to *Editing the log_hosts.config File, on page 257*).

When ICP and HTTP host log splitting are all enabled, Traffic Edge generates separate log files for HTTP/FTP transactions, based on the origin server, and places all ICP transactions in their own respective log files; for example, if the `log_hosts.config` file contains the two origin servers `uni.edu` and `company.com`, and the Squid format is enabled. Traffic Edge generates the following log files.

Log Filename	Description
`squid-uni.edu.log`	All HTTP and FTP transactions for `uni.edu`
`squid-company.com.log`	All HTTP and FTP transactions for `company.com`
`squid-icp.log`	All ICP transactions for all hosts
`squid.log`	All HTTP and FTP transactions for other hosts

If you disable ICP log splitting, ICP transactions are placed in the same log file as HTTP and FTP transactions. Using the previous example hosts and assuming the Squid log format is used, Traffic Edge generates these log files:

Log Filename	Description
`squid-uni.edu.log`	All entries for `uni.edu`
`squid-company.com.log`	All entries for `company.com`
`squid.log`	All other entries

Traffic Edge also enables you to create XML-based custom log formats that offer even greater control over log file generation based on protocol and hostname. For more information, refer to *Using the Custom Format, on page 246*.

Setting Log Splitting Options

You can set log splitting options either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ To set log splitting options from Traffic Manager:

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Subsystems** button and then click the **Logging** button.
3. Click the **Splitting** tab.
4. Enable the **Split ICP Logs** option to record all ICP transactions in a separate log file. Disable the **Split ICP Logs** option to record all ICP transactions in the same log file as HTTP/FTP transactions.
5. Enable the **Split Host Logs** option to record all HTTP and FTP transactions for each origin server listed in `log_hosts.config` in a separate log file. Disable the **Split Host Logs** option to record all HTTP and FTP transactions for each origin server listed in `log_hosts.config` in the same log file.
6. Click the **Apply** button.

▼ To set log splitting options manually:

1. In a text editor, open the `records.config` file located in the `config` directory.
2. Edit the following variables:

Variable	Description
proxy.config.log2.separate_icp_logs	Set this variable to 1 to record all ICP transactions in a separate log file. Set this variable to 0 to record all ICP transactions in the same log file as HTTP and FTP transactions. Set this variable to -1 to filter all ICP transactions from the standard log files.
proxy.config.log2.separate_host_logs	Set this variable to 1 to record HTTP and FTP transactions for each host listed in log_hosts.config file in a separate log file. Set this variable to 0 to record all HTTP and FTP transactions for each host listed in the log_hosts.config file in the same log file.

3 Save and close the `records.config` file.

4 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

5 Run the command `traffic_line -x` to apply the configuration changes.

Editing the log_hosts.config File

The default `log_hosts.config` file is located in the Traffic Edge `config` directory. To record HTTP and FTP transactions for different origin servers in separate log files, you must specify the hostname of each origin server on a separate line in the file.

Tip You can specify keywords in the `log_hosts.config` file to record in a separate log file all transactions from origin servers that contain the specified keyword in their names; for example, if you specify the keyword `sports`, Traffic Edge records all HTTP and FTP transactions from `sports.yahoo.com` and `www.foxsports.com` in a log file called `squid-sports.log` (if the Squid format is enabled).

Note If Traffic Edge is clustered and if you enable log file collation, Inktomi recommends that you use the same `log_hosts.config` file on every Traffic Edge node in the cluster.

▼ **To edit the log_hosts.config file:**

1 In a text editor, open the `log_hosts.config` file located in the Traffic Edge `config` directory.

2 Enter the hostname of each origin server on a separate line in the file: for example,

```
webserver1
webserver2
webserver3
```

3 Save and close the `log_hosts.config` file.

4 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

5 Run the command `traffic_line -x` to apply the configuration changes.

Collating Event Log Files

You can use the Traffic Edge log file collation feature to keep all logged information in one place. Log collation allows you to analyze a set of Traffic Edge clustered nodes as a whole rather than as individual nodes and to use a large disk that might only be located on one of the nodes in the cluster.

Traffic Edge collates log files by using one or more nodes as log collation servers and all remaining nodes as log collation clients. When a Traffic Edge node generates a buffer of event log entries, it determines whether it is the collation server or a collation client. The collation server node writes all log buffers to its local disk, just as it would if log collation were not enabled. Log collation servers can be standalone or they can be part of a node running Traffic Edge.

The collation client nodes prepare their log buffers for transfer across the network and send the buffers to the log collation server. When the log collation server receives a log buffer from a client, it writes it to its own log file as if it were generated locally; see *Figure 37*.

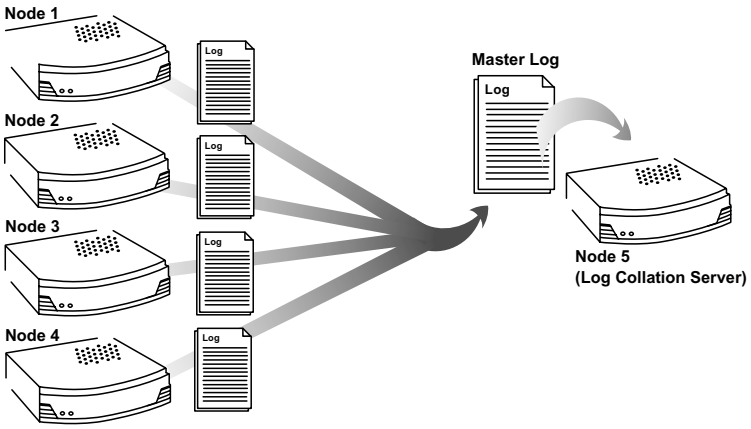


Figure 37 Log collation

If log clients cannot contact their log collation server, they write their log buffers to their local disks, into *orphan* log files. Orphan log files require manual collation.

Note Log collation can have an impact on network performance. Because all nodes are forwarding their log data buffers to the single collation server, a bottleneck can occur.

Note Collated log files contain timestamp information for each entry, but entries do not appear in the files in strict chronological order. You can sort collated log files before doing analysis.

To configure Traffic Edge to collate event log files, you must perform the following tasks:

- Either configure a Traffic Edge node to be a collation server *or* install and configure a standalone collator; refer to *Configuring Traffic Edge to Be a Collation Server*, below or refer to *Using a Standalone Collator, on page 259*.
- Configure Traffic Edge nodes to be collation clients; refer to *Configuring Traffic Edge to Be a Collation Client, on page 260*.
- Add an attribute to the `LogObject` specification in the `logs_xml.config` file if you are using custom log file formats; refer to *Collating Custom Event Log Files, on page 262*.

  By default, Traffic Edge uses custom log file formats to record streaming media transactions. Make sure you edit the `logs_xml.config` file as described in *Collating Custom Event Log Files, on page 262* if you are serving streaming media.

Configuring Traffic Edge to Be a Collation Server

You can configure a Traffic Edge node to be a collation server either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ **To configure Traffic Edge to be a collation server from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Subsystems** button and then click the **Logging** button.
3. Click the **Collation** tab.
4. In the **Collation Mode** section, enable the **Be A Collation Server** option.
5. In the **Log Collation Port** field, enter the port number used for communication with collation clients. The default port number is 8085.
6. In the **Log Collation Secret** field, enter the password used to validate logging data and prevent the exchange of arbitrary information.

Note All collation clients must use this same secret.

7. Click the **Apply** button.

▼ **To configure Traffic Edge to be a collation server manually:**

1. In a text editor, open the `records.config` file located in the `config` directory.
2. Edit the following variables:

Variable	Description
proxy.config.log2.collation_mode	Set this variable to 1 to set this Traffic Edge node as a log collation server.
proxy.config.log2.collation_port	Set this variable to specify the port number used for communication with collation clients. The default port number is 8085.
proxy.config.log2.collation_secret	Set this variable to specify the password used to validate logging data and prevent the exchange of arbitrary information. All collation clients must use this same secret.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

IMPORTANT If you modify the collation port or secret after connections between the collation server and collation clients have been established, you must restart Traffic Edge.

Using a Standalone Collator

If you do not want the log collation server to be a Traffic Edge node, you can install and configure a standalone collator (SAC), which can dedicate more of its power to collecting, processing, and writing log files.

Note The standalone collator is currently available for the UNIX platform only.

▼ **To install and configure a standalone collator:**

1. Configure your Traffic Edge nodes as log collation clients; refer to *Configuring Traffic Edge to Be a Collation Client*, below.
2. Copy the `sac` binary from the Traffic Edge `bin` directory to the machine serving as the standalone collator.
3. Create a directory called `config` in the directory that contains the `sac` binary.
4. Create a directory called *`internal`* in the `config` directory you created in *step 3*. This directory is used internally by the standalone collator to store lock files.
5. Copy the `records.config` file from a Traffic Edge node configured to be a log collation client to the `config` directory you created in *step 3* on the standalone collator.

   The `records.config` file contains the log collation secret and port you specified when configuring Traffic Edge nodes to be collation clients. The collation port and secret must be the same for all collation clients and servers.
6. In a text editor, open the `records.config` file on the standalone collator and edit the following variable:

Variable	Description
proxy.config.log2.logfile_dir	Set this variable to specify the directory on which you want to store the log files. You can specify an absolute path to the directory or a path relative to the directory from which the `sac` binary is executed. Note: The directory must already exist on the machine serving as the standalone collator.

7. Save and close the `records.config` file.
8. Enter the following command:

   ```
   sac -c config
   ```

Configuring Traffic Edge to Be a Collation Client

You can configure a Traffic Edge node to be a collation client either by using Traffic Manager or by editing a configuration file manually. Both procedures are provided below.

▼ **To configure Traffic Edge to be a collation client from Traffic Manager:**

1. From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.
2. On the **Configure** tab, click the **Subsystems** button and then click the **Logging** button.
3. Click the **Collation** tab.
4. In the **Collation Mode** section, enable the **Be a Collation Client** option to set the Traffic Edge node as a collation client and send the active standard formatted log entries (such as Squid and Netscape) to the log collation server.

Note To send custom XML-based formatted log entries to the collation server, you must add a log object specification to the `logs_xml.config` file; refer to *Using the Custom Format, on page 246*.

5 In the **To Collation Server** field, enter the hostname of the collation server. This could be the Traffic Edge collation server or a standalone collation server.

6 In the **Log Collation Port** field, enter the port number used for communication with the collation server. The default port number is 8085.

7 In the **Log Collation Secret** field, enter the password used to validate logging data and prevent the exchange of arbitrary information. This must be the same secret you set on the collation server.

8 Enable the **Log Collation Host Tagged** option if you want to preserve the origin of log entries in the collated log files.

9 In the **Log Collation Orphan Space** field, enter the maximum amount of space (in megabytes) you want to allocate to the logging directory on the collation client for storing orphan log files. (Orphan log files are created when the log collation server cannot be contacted). The default value is 25 MB.

10 Click the **Apply** button.

▼ To configure Traffic Edge node to be a log collation client manually:

1 In a text editor, open the `records.config` file located in the `config` directory.

2 Edit the following variables:

Variable	Description
proxy.config.log2.collation_mode	Set this variable to 2 to configure this Traffic Edge node to be a log collation client and send standard formatted log entries to the collation server. To send custom XML-based formatted log entries to the collation server, you must add a log object specification to the logs_xml.config file; refer to *Using the Custom Format, on page 246*.
proxy.config.log2.collation_host	Specify the hostname of the collation server.
proxy.config.log2.collation_port	Specify the port used for communication with the collation server. The default port number is 8085.
proxy.config.log2.collation_secret	Specify the password used to validate logging data and prevent the exchange of arbitrary information.

Variable	Description
proxy.config.log2.collation_host_tagged	Set this variable to 1 if you want the hostname of the collation client that generated the log entry to be included in each entry. Set this variable to 0 if you do *not* want the hostname of the collation client that generated the log entry to be included in each entry.
proxy.config.log2.max_space_mb_for_orphan_logs	Set this variable to specify the maximum amount of space (in megabytes) you want to allocate to the logging directory on the collation client for storing orphan log files. Orphan log files are created when the log collation server cannot be contacted. The default value is 25 MB.

3 Save and close the `records.config` file.

4 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

5 Run the command `traffic_line -x` to apply the configuration changes.

IMPORTANT If you modify the collation port or secret after connections between the collation clients and the collation server have been established, you must restart Traffic Edge.

Collating Custom Event Log Files

If you use custom event log files, you must edit the `logs_xml.config` file in addition to configuring a collation server and collation clients. Use the following procedure.

▼ **To collate custom event log files:**

1 On each collation client, open the `logs_xml.config` file in a text editor (located in the Traffic Edge `config` directory).

2 Add the `CollationHosts` attribute to the `LogObject` specification, as shown below:

```
<LogObject>
   <Format = "squid"/>
   <Filename = "squid"/>
   <CollationHosts="ipaddress:port"/>
</LogObject>
```

ipaddress is the hostname or IP address of the collation server to which all log entries (for this object) are forwarded. *port* is the port number on which the collation server communicates with collation clients.

For streaming media, add the `CollationHosts` attribute to the `LogObject` specification in the Media-IXT summary log format and/or the Media-IXT complete log format at the end of the file, as shown below:

```
<!-- Media-IXT summary log object -->
<LogObject>
    <Format="summary-mixt"/>
    <Filename="summary-mixt"/>
    <Protocols="mixt"/>
    <CollationHosts="ipaddress:port"/>
</LogObject>

<!-- Media-IXT complete log object -->
<LogObject>
   <Format="complete-mixt"/>
   <Filename="complete-mixt"/>
   <Protocols="mixt"/>
   <CollationHosts="ipaddress:port"/>
</LogObject>
```

3 Save and close the `logs_xml.config` file.

4 In UNIX, navigate to the Traffic Edge `bin` directory.

In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

5 Run the command `traffic_line -L` to restart Traffic Edge on the local node or `traffic_line -M` to restart Traffic Edge on all the nodes in a cluster.

Working with Streaming Media Log Files

Traffic Edge collects log information about on-demand and live streams for all supported media types. By default, Traffic Edge creates the following log files:

- `complete-mixt.log`, which includes all of the information gathered by Traffic Edge for streaming media
- `summary-mixt.log`, which provides a summary of the information gathered by Traffic Edge for streaming media

The streaming media log files are custom log files. You configure them by editing the `logs_xml.config` file, located in the Traffic Edge `config` directory.

By default, Traffic Edge also creates a log file in standard Squid format for all nonstreaming media transactions (HTTP and FTP), in addition to the `complete-mixt.log` and `summary-mixt.log` files. However, if Traffic Edge does not process nonstreaming media transactions, it does not create Squid log files.

QuickTime Logging

For QuickTime, log entries appear after the QuickTime Player stops the connection to Traffic Edge (a minimum of 30 seconds after the stream has finished playing). Traffic Edge downloads as much as needed from a QuickTime origin server and then it switches to playing from the cache.

Because live streams are not cached, the `prcb` field for QuickTime is always `0` for live streams.

In hierarchical live-splitting deployments, QuickTime log collection has the following characteristics:

- The origin QuickTime server packetizes each live stream according to the player bandwidth settings. For each client connecting at a different speed, Traffic Edge obtains another instance of a given live stream from the origin QuickTime server. Each instance is called a *live splitter*. Traffic Edge maintains a table of the live splitters it has open at any given time.
- The suffix `.sdp` typically indicates a live stream.

Traffic Edge takes the following actions when a QuickTime Player requests a URL:

1. Traffic Edge opens a connection to the origin QuickTime server, obtaining metadata that indicates if the stream is live or on-demand.
2. If the stream is live, Traffic Edge looks in the live splitter table:
   - If the requested stream matches a live splitter, Traffic Edge attaches the client to the appropriate live splitter after closing the connection it used to obtain metadata.
   - If the requested stream does not match a live splitter, Traffic Edge closes the connection it used to obtain metadata and then opens another, setting up a new live splitter for the client.

In hierarchical live splitting, a Traffic Edge node connected to a QuickTime Player only writes a log entry if the player sends a `PLAY` command. A parent Traffic Edge node connected to a child Traffic Edge node only writes a log entry if the child relays a `PLAY`

command after receiving one from the player. The log entry on the parent node is written when the last child node disconnects.

Only one entry per live split stream is logged on the parent, while as many entries as there are client players for that stream are logged on the child Traffic Edge node.

Real Networks Logging

The RealProxy component of Traffic Edge maintains the `proxy.log` file, which records both live and on-demand Real Networks transactions. Traffic Edge records both on-demand and live Real Networks transactions in the `complete-mixt.log` and `summary-mixt.log` files.

In hierarchical deployments, the RealProxy `proxy.log` file and the Traffic Edge `summary-mixt.log` and `complete-mixt.log` files handle the same connections differently.

For on-demand streams:

- The Real Proxy `proxy.log` file on a child cache shows `Demand Cache Hit`. On the parent cache, the `proxy.log` file shows `demand passthrough`.
- The Traffic Edge `summary-mixt.log` and `complete-mixt.log` files show a dash (-) in the `styp` field.

For live streams:

- The RealProxy `proxy.log` file on a child cache shows `live split` (provided the origin RealServer allows splitting for that content). On the parent cache, the `proxy.log` file shows `live passthrough`.

Refer to the RealProxy documentation for Real Networks definitions of demand cache hit, demand passthrough, live split, and live passthrough.

In a hierarchical deployment where both Traffic Edge nodes are in forward proxy mode, two log entries appear in the parent cache RealProxy `proxy.log` file for live streams. One entry shows port 554 and 0 bytes; this is for a connection that serves an internal purpose for RealProxy. The other entry shows port 3030 (the data connection) and provides accurate information about the client connection.

The RealProxy `proxy.log` file on the child is not affected.

The client_GUID Field in RealProxy and RealServer Log Files

Origin RealServers maintain a log file called `rmaccess.log`. The `client_GUID` logging field appears both in the `rmaccess.log` file and in the RealProxy component of the Traffic Edge `proxy.log` file. The `client_GUID` field records a unique user ID for every Real media player. Because of privacy concerns, this field only records a meaningful value if the Real media player user enables the **Send RealPlayer GUID to RealServers** option.

If the **Send RealPlayer GUID to RealServers** option in the Real media player is disabled (the default value):

- The RealProxy logs all 0s in the `client_GUID` field of the `proxy.log` file.
- The origin RealServer logs random, unreliable information in the `client_GUID` field of the `rmaccess.log` file.

If the **Send RealPlayer GUID to RealServers** option in the Real media player is enabled:

- The RealProxy logs the correct value in the `client_GUID` field of the `proxy.log` file.

- The origin RealServer logs the correct value in the `client_GUID` field of the `rmaccess.log` file.

To enable the **Send RealPlayer GUID to RealServers** option on the Real media player, select **Preferences** from the **View** menu and then click the **Support** tab. Check the **Send RealPlayer GUID to RealServers** box.

The **Send connection-quality data to RealServers** option in the **Preferences** menu does not affect the `client_GUID` logging field.

SMIL Files

An SMIL (Synchronized Multimedia Integration Language) file is an HTTP-like text file that is tailored for Real Networks streaming. SMIL files invoke other files, including both streaming media files (such as movie files) and various types of multimedia files (such as JPEG or GIF image files). SMIL files specify how and when the invoked files should play.

SMIL files typically refer to one or more files with the `.rp` extension. The `.rp` file is located between the SMIL file and the invoked content files: one SMIL file might reference several `.rp` files and one `.rp` file might reference several JPEG or GIF files.

Traffic Edge SMIL logging works as follows:

- The SMIL file itself shows a relatively small size (SMIL files are lightweight compared to an image or sound file).
- The files invoked by the SMIL file have their own log entries, showing their individual byte counts.
- Some files invoked by the SMIL file might be logged as zero bytes long.

RealProxy has its own style of SMIL logging in the `proxy.log` file. The files invoked by SMIL files do not have their own log entries, except for the `.rp` file, whose byte counts approximately equal the total byte count of the JPEG, GIF, and other content files that it invokes.

Viewing Logging Statistics

Traffic Edge generates logging statistics that help you see the following information:

- How many log files (formats) are currently being written.
- The current amount of space being used by the logging directory, which contains all of the event and error logs.
- The number of access events written to log files since Traffic Edge installation. This counter represents one entry in one file. If multiple formats are being written, a single event creates multiple event log entries.
- The number of access events skipped (because they were filtered) since Traffic Edge installation.
- The number of access events that have been written to the event error log since Traffic Edge installation.

You can view the statistics from the **Monitor** tab in Traffic Manager or retrieve them using the Traffic Line command-line interface; refer to *Chapter 12, Monitoring Traffic*.

Viewing Log Files

You can view the system, event, and error log files that Traffic Edge creates from Traffic Manager. You can view an entire log file, a specified last number of lines in the log file, or all lines that contain a specified string.

In addition to viewing a log file, you can delete a log file or copy it to your local system.

Note You must have the correct user permissions to copy and delete log files.

Traffic Edge displays only one MB of information in the log file. If the log file you select to view is bigger than one MB, Traffic Edge truncates the file and displays a warning message indicating that the file is too big.

To view, copy, or delete a system, event, or error log file, use the following procedure.

▼ **To view, copy, or delete a log file:**

1 From your browser, access Traffic Manager; refer to *Accessing Traffic Manager, on page 27*.

2 On the **Configure** tab, click the **Logs** button under **My Proxy**.

The **System** tab displays.

3 To view, copy, or delete a system log file, go to *step 4*.

To view, copy, or delete an event or error log file, click the **Access** tab.

4 In the **Log File** drop-down list, select the log file you want to view, copy, or delete.

Traffic Edge lists the system log files logged with the system-wide logging facility `syslog` under the daemon facility.

Traffic Edge lists the event log files located in the directory specified in the **Logging Directory** field under **Subsystems**/**Logging** in Traffic Manager or by the configuration variable `proxy.config.log2.logfile_dir` in the re`cords.config` file. The default directory is `logs` in the Traffic Edge installation directory.

5 In the **Action** area, select one of the following options:

- **Display the selected log file** to display the entire log file.
- **Display last lines of the selected file** to display the last lines of the log file. Enter the number of lines you want to view in the field provided.
- **Display lines that match in the selected log file** to display all the lines in the log file that match a particular string. Enter the string in the field provided.
- **Remove the selected log file** to delete the selected log file from the Traffic Edge system.
- **Save the selected log file in local filesystem** to save a copy of the selected log file on your local system.

6 Click the **Apply** button.

If you selected to view the log file, Traffic Edge displays the file at the end of the page.

If you selected to delete the log file, Traffic Edge deletes the file. You are not prompted to confirm the deletion.

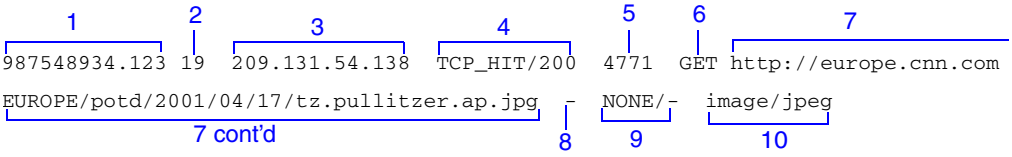
If you selected to save the log file, you are prompted for the location in which you want to save the file on your local system.

Example Event Log File Entries

This section shows an example log file entry in each of the standard log formats supported by Traffic Edge: Squid, Netscape Common, Netscape Extended, and Netscape Extended-2. *Streaming Media Log Files, on page 272* shows example streaming media log file entries.

Squid Format

The following figure shows a sample log entry in a `squid.log` file.

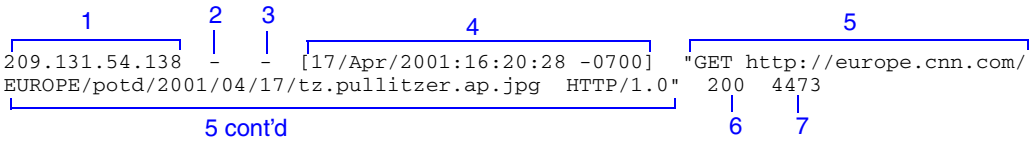


The following table describes each field.

Field	Symbol	Description
1	cqtq	The client request timestamp in Squid format; the time of the client request in seconds since January 1, 1970 UTC (with millisecond resolution).
2	ttms	The time Traffic Edge spent processing the client request; the number of milliseconds between the time that the client established the connection with Traffic Edge and the time that Traffic Edge sent the last byte of the response back to the client.
3	chi	The IP address of the client's host machine.
4	crc/pssc	The cache result code; how the cache responded to the request: HIT, MISS, and so on. Cache result codes are described on *page 476*. The proxy response status code (the HTTP response status code from Traffic Edge to client).
5	psql	The length of the Traffic Edge response to the client in bytes, including headers and content.
6	cqhm	The client request method: GET, POST, and so on.
7	cquc	The client request canonical URL; blanks and other characters that might not be parsed by log analysis tools are replaced by escape sequences. The escape sequence is a percentage sign followed by the ASCII code number of the replaced character in hex.
8	caun	The username of the authenticated client. A hyphen (-) means that no authentication was required.
9	phr/pqsn	The proxy hierarchy route; the route Traffic Edge used to retrieve the object. The proxy request server name; the name of the server that fulfilled the request. If the request was a cache hit, this field contains a hyphen (-).
10	psct	The proxy response content type; the object content type taken from the Traffic Edge response header.

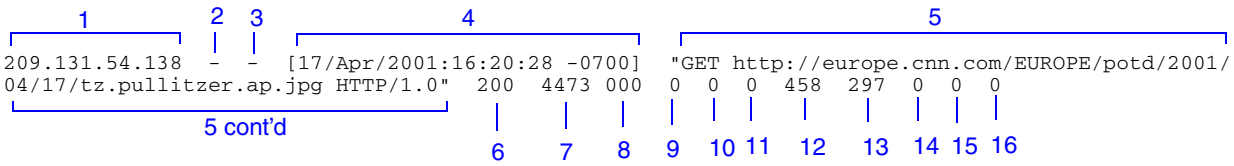
Netscape Common

The following figure shows a sample log entry in a `common.log` file.



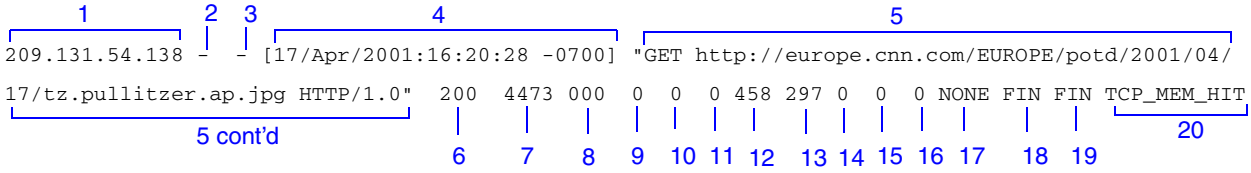
Netscape Extended

The following figure shows a sample log entry in an `extended.log` file.



Netscape Extended-2

The following figure shows a sample log entry in an `extended2.log` file.



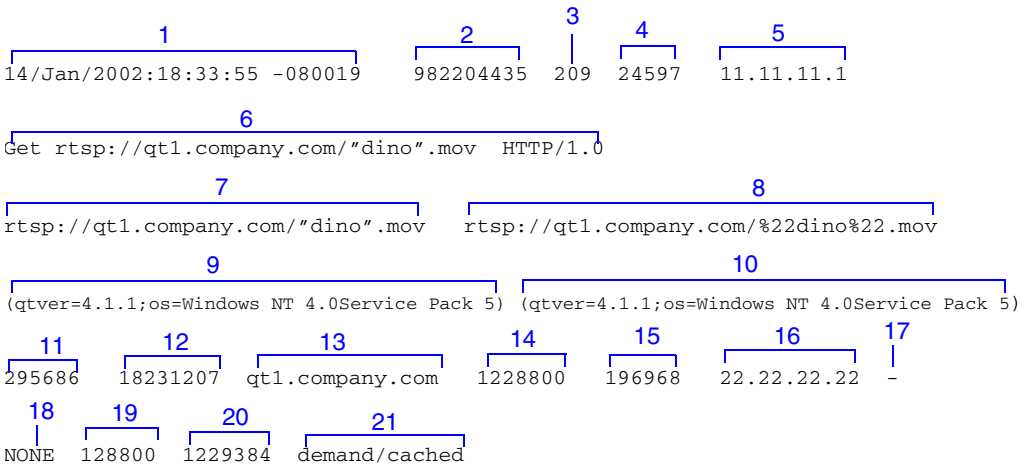
The following table describes each field.

Field	Symbol	Description
		Netscape Common
1	chi	The IP address of the client's host machine.
2		This hyphen (-) is always present in Netscape log entries.
3	caun	The authenticated client username. A hyphen (-) means no authentication was required.
4	cqtd	The date and time of the client request, enclosed in brackets.
5	cqtx	The request line, enclosed in quotes.
6	pssc	The proxy response status code (HTTP reply code).
7	pscl	The length of the Traffic Edge response to the client in bytes.
		Netscape Extended
8	sssc	The origin server response status code.
9	sshl	The server response transfer length; the body length in the origin server response to Traffic Edge, in bytes.

Field	Symbol	Description
10	cqbl	The client request transfer length; the body length in the client request to Traffic Edge, in bytes.
11	pqbl	The proxy request transfer length; the body length in the Traffic Edge request to the origin server.
12	cqhl	The client request header length; the header length in the client request to Traffic Edge.
13	pshl	The proxy response header length; the header length in the Traffic Edge response to the client.
14	pqhl	The proxy request header length; the header length in Traffic Edge request to the origin server.
15	sshl	The server response header length; the header length in the origin server response to Traffic Edge.
16	tts	The time Traffic Edge spent processing the client request; the number of seconds between the time that the client established the connection with Traffic Edge and the time that Traffic Edge sent the last byte of the response back to the client.
Netscape Extended2		
17	phr	The proxy hierarchy route; the route Traffic Edge used to retrieve the object.
18	cfsc	The client finish status code: FIN if the client request completed successfully or INTR if the client request was interrupted.
19	pfsc	The proxy finish status code: `FIN` if the Traffic Edge request to the origin server completed successfully or `INTR` if the request was interrupted.
20	crc	The cache result code; how the Traffic Edge cache responded to the request: HIT, MISS, and so on. Cache result codes are described on *page 476*.

Streaming Media Log Files

The following figure shows a sample log entry in the default `complete-mixt.log` file.



The following table describes each field.

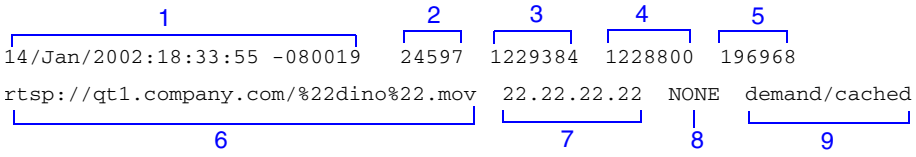
Field	Symbol	Description
1	`cqtn`	The date and time of the client request, in Netscape timestamp format.
2	`cqts`	The client request timestamp in Squid format; the time of the client request in seconds since January 1, 1970 UTC (with millisecond resolution).
3	`tfcb`	The time to first client byte in milliseconds. Used for quality of service logging.
4	`ttms`	The time Traffic Edge spent processing the client request; the number of milliseconds between the time that the client established the connection with Traffic Edge and the time that Traffic Edge returned the last byte of the response to the client.
5	`chi`	The IP address of the client's host machine. For PNA requests, the value for this field is always 0. If the request is from a RealOne player and the RTSP proxy port is set to the default port 554 either explicitly or transparently, this field shows the IP address of Traffic Edge instead of the IP address of the client.
6	`cqtx`	The full text of the HTTP client request, minus headers. In reverse proxy mode, Traffic Edge logs the rewritten (mapped) URL (according to the rules in the `remap.config` file), *not* the pristine (unmapped) URL. To configure Traffic Edge to log the original, unmapped URL, set the variable `proxy.config.url_remap.pristine_host_hdr` in the `records.config` file to 1.

Field	Symbol	Description
7	cqu	The universal resource identifier (URI) of the request from client to Traffic Edge. In reverse proxy mode, Traffic Edge logs the rewritten (mapped) URL (according to the rules in the remap.config file), *not* the pristine (unmapped) URL. To configure Traffic Edge to log the original, unmapped URL, set the variable proxy.config.url_remap.pristine_host_hdr in the records.config file to 1.
8	cquc	The client request canonical URL; differs from cqu in that blanks (and other characters that might not be parsed by log analysis tools) are replaced by escape sequences. The escape sequence is a percentage sign followed by the ASCII code number in hex. In reverse proxy mode, Traffic Edge logs the rewritten (mapped) URL (according to the rules in the remap.config file), *not* the pristine (unmapped) URL. To configure Traffic Edge to log the original, unmapped URL, set the variable proxy.config.url_remap.pristine_host_hdr in the records.config file to 1.
9	cgid	The client GUID (global unique identifier); a special string that the client sends to the RealServer, indicating that it is registering a request.
10	caun	The client authenticated username; result of the RFC931/ident lookup of the client username. If the authenticated username is not available, the GUID is used (this is always the case for Real Networks).
11	band	The bandwidth of data sent to the client player, in bits per second.
12	fsiz	The size of the file, in bytes, as seen by the origin server. Windows Media Player sees a smaller size in the case of multibitrate files.
13	shn	The hostname of the origin server.
14	prcb	The number of proxy response bytes to the client from the cache. This value is always 0 for live streams.
15	prob	The number of proxy response bytes to the client from the origin server. For QuickTime, the prob field is always a nonzero number on cache hits. Traffic Edge tracks both the control and data bytes obtained from the QuickTime origin server. The control bytes are the product of RTSP control commands such as DESCRIBE, SETUP, and PLAY, and are typically 2 to 3 kilobytes per transaction. These values result in a nonzero value for the prob field even when streaming media data is served from the cache.
16	pqsi	The proxy request server IP address (the IP address of the parent for requests to parent proxies).
17	pqsn	The proxy request server name.

Field	Symbol	Description
18	`phr`	The proxy hierarchy route; the route that Traffic Edge used to retrieve the object: `phr` can one of the following values: `NONE`. Access to the stream is denied; for example, a `filter.config` file rule blocks access to the stream. `DIRECT`. Traffic Edge is connecting to the origin server without going through a parent. `DEFAULT_PARENT`. Traffic Edge is connecting to a parent (configured through the parent.config file) to retrieve the content (QuickTime and Real Networks only). `FIRST_UP_PARENT`. Traffic Edge is connecting to the first available parent (WMT only). When Traffic Edge serves a WMT stream from the cache, a small number of bytes are logged as coming from the origin server. If the client uses Windows Media Player Version 6, more bytes are logged than if the client is using Windows Media Player Version 7. This is the result of Windows Media Player behavior.
19	`pscl`	The proxy response transfer length; the length of the Traffic Edge response to the client, in bytes.

Field	Symbol	Description
20	`psql`	The proxy response transfer length, in Squid format (includes header and content length).
21	`styp`	*streamtype* can have the following values: ◆ `cached` - the stream was written to and read from the cache but was not authenticated by the origin server. ◆ `cached-origin-authenticated` - the stream was written to and read from the cache and was authenticated by the origin server. ◆ `passthrough` - the stream was proxied in passthrough mode. Traffic Edge does not cache it. ◆ `passthrough-filter` - the stream was proxied in passthrough mode because the `cache.config` file rules specify that it must not be cached. ◆ `passthrough-origin-authenticated` - the stream was proxied in passthrough mode because the content is authenticated. ◆ `split` - the stream was split from the origin server (client-side log). ◆ `split-origin` - the stream was split from the origin server (origin server log). ◆ `denied-filter` - the request was denied according to the rules in the `filter.config` file. ◆ `denied-origin-auth-failure` - the origin authentication challenge failed. ◆ `denied-proxy-auth-failure` - the proxy authentication challenge failed. ◆ `denied-proxy` - the request was denied by Traffic Edge; for example, memory throttling or bandwidth throttling ocurred. ◆ `denied-origin` - the request was denied at the parent Traffic Edge or at the origin server for reasons other than authentication failure. ◆ `non-existent` - the stream does not exist on the origin server. When Traffic Edge serves authenticated QuickTime streams, it proxies but does not cache the streams. Traffic Edge logs `passthrough--origin-authenticated` for this type of transaction.

The following figure shows a sample log entry in the default `summary-mixt.log` file.



The following table describes each field.

Field	Symbol	Description
1	`cqtn`	Refer to *`cqtn`, on page 272*.
2	`ttms`	Refer to *`ttms`, on page 272*.
3	`psql`	Refer to *`psql`, on page 275*.
4	`prcb`	Refer to *`prcb`, on page 273*.
5	`prob`	Refer to *`prob`, on page 273*.
6	`cquc`	Refer to *`cquc`, on page 273*.
7	`pqsi`	Refer to *`pqsi`, on page 273*.
8	`phr`	Refer *`phr`, on page 274*.
9	`styp`	Refer to *`styp`, on page 275*.

Support for Traditional Custom Logging

Previous Traffic Edge releases provide traditional custom logging in addition to the XML-based custom logging. Although this release of Traffic Edge continues to support traditional custom logging, Inktomi recommends that you use the XML-based custom logging, which is more versatile.

If you have upgraded Traffic Edge from a previous release, the installation program configures Traffic Edge to use XML-based custom logging instead of the traditional custom logging. The installation program automatically converts your `logs.config` file to a `logs_xml.config` file using the format converter `cust_log_fmt_cnvrt`. Traffic Edge retains your original `logs.config` file in the Traffic Edge `config` directory so that you can use your traditional log formats, if you prefer.

The format converter only converts traditional log configuration files named `logs.config`. If you are using a traditional log configuration file with a name other than `logs.config`, you must convert the file yourself after installation; refer to *Using cust_log_fmt_cnvrt, on page 278*.

If you opt to use traditional custom logging instead of the more versatile XML-based custom logging, you must enable the traditional custom logging option manually. In addition, if you want to configure Traffic Edge as a collation client that sends log entries in traditional custom formats, you must set collation options manually. Use the following procedures.

Enabling Traditional Custom Logging

To enable custom logging, you must edit a configuration file manually.

▼ **To enable traditional custom logging:**

1. In a text editor, open the `records.config` file located in the `config` directory.
2. Edit the following variables:

Variable	Description
proxy.config.log2.custom_logs_enabled	Set this variable to 1 to enable custom logging.
proxy.config.log2.xml_logs_config	Set this variable to 0 to disable XML-based custom logging.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

To edit your existing traditional custom log formats, modify the `logs.config` file as before; refer to *logs.config, on page 385*.

To configure your Traffic Edge node to be a collation client and send traditional custom log files to the collation server, use the following procedure.

▼ **To configure Traffic Edge as a collation client:**

1. In a text editor, open the `records.config` file located in the `config` directory.
2. Edit the following variables:

Variable	Description
proxy.config.log2.collation_mode	Set this variable to 3 to configure this Traffic Edge node to be a log collation client and send log entries in the traditional custom formats to the collation server. Set this variable to 4 to configure this Traffic Edge node to be a log collation client and send log entries in both the standard formats (Squid, Netscape) and the traditional custom formats to the collation server.
proxy.config.log2.collation_host	Specify the hostname of the collation server.
proxy.config.log2.collation_port	Specify the port Traffic Edge uses to communicate with the collation server. The default port number is 8085.
proxy.config.log2.collation_secret	Specify the password used to validate logging data and prevent the exchange of arbitrary information.
proxy.config.log2.collation_host_tagged	Set this variable to 1 if you want the hostname of the collation client that generated the log entry to be included in each entry. Set this variable to 0 if you do *not* want the hostname of the collation client that generated the log entry to be included in each entry.

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

Using cust_log_fmt_cnvrt

The format converter `cust_log_fmt_cnvrt` converts your traditional custom log configuration file (`logs.config`) to an XML-based custom log configuration file (`logs_xml.config`) so that you can use the Traffic Edge XML-based custom logging.

▼ **To run the format converter:**

1. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.

2 Enter the command `cust_log_fmt_cnvrt` and include the options you want to use.

The format of the command is

`cust_log_fmt_cnvrt [-o output_file | -a] [-hnVw] [input_file..]`

The following table describes the command-line options.

Option	Description
`-o output_file`	Specifies the name of the output file. You can specify one output file only. If you specify multiple input files, the converter combines the converted output from all the files into a single output file. This option and the `-a` option are mutually exclusive. If you want to create multiple output files from multiple input files, you must use the -a option. If you do not specify an output file (using the `-o` or `-a` options), output goes to `stdout`.
`-a`	Generates one output file for each input file. The format converter creates the name of the output file automatically from the name of the input file by replacing `.config` at the end of the filename with `_xml.config`. Note: If the source filename does not contain a `.config` extension, the converter appends `_xml.config` to the source filename to form the new filename.
`-h`	Displays a description of the `cust_log_fmt_cnvrt` options.
`-n`	Annotates the output file(s) with comments about the success or failure of the translation process for each of the input lines. This option produces a comment at the beginning of the output file(s) that describes any error that the format converter encountered while converting the file. The comment includes the line number, the input line type (format, filter, or unknown), and either a success status or a description of the error encountered.
`-V`	Displays the version of the format converter you are running.
`-w`	Overwrites existing output files without warning. If you do not specify the `-w` option, the format converter does not overwrite existing output files; if you specify an output file that already exists, the converter does not convert the input file.
input file	Specifies the name of the input file. If you do not specify an input filename, the format converter takes the input from `stdin`.

Examples

The following example converts the file `logs.config` and sends the results to `stdout`:

```
cust_log_fmt_cnvrt logs.config
```

The following example converts a `logs.config` file into a `logs_xml.config` file and annotates the output file (`logs_xml.config`) with comments about the success or failure of the translation process. If a file named `logs_xml.config` already exists, the format converter overwrites it.

```
cust_log_fmt_cnvrt -o logs_xml.config -n -w logs.config
```

The following example converts the files `x.config`, `y.config`, and `z.config` into three separate output files called `x_xml.config`, `y_xml.config`, and `z_xml.config`:

```
cust_log_fmt_cnvrt -a x.config y.config z.config
```

Understanding the Antivirus Extension Log Files

The Antivirus Extension generates two log files in the Traffic Edge `logs` directory: `vscan.log` and `vscan_stats.log`.

Viewing the vscan.log File

The `vscan.log` file records one of the following entries for each object that Traffic Edge sends to CarrierScan Server for scanning:

- `CLEAN`

  CarrierScan Server did not find any viruses.

- `INFECTED_REPAIRED`

  CarrierScan Server found a virus and repaired the file.

- `INFECTED_PARTIALLY_REPAIRED`

  CarrierScan Server found a virus that it cannot fully repair. Traffic Edge sends an error code to the user and denies the user access.

- `INFECTED_NOT_REPAIRED`

  CarrierScan Server found a virus that it cannot repair. Traffic Edge sends an error to the client, sends an error code to the user, and denies the user access.

- `SCSCANSERVER_ERROR`

  CarrierScan Server returned a system error. Traffic Edge sends an error to the client, sends an error code to the user, and denies the user access.

Viewing the vscan_stats.log File

The `vscan_stats.log` file displays the following information:

- The total number of scanning transactions
- The average total scanning time per object, in milliseconds
- The average CarrierScan Server scanning time
- The average size of the objects scanned, in kilobytes
- The total amount of data sent for scanning, in kilobytes
- The total number of objects scanned and found to be clean
- The total number of objects scanned and found to be infected
- The total number of objects scanned, found to be infected, and repaired
- The total number of scanning errors returned from CarrierScan Server
- The CarrierScan Server timeout
- The number of scanning request overloads
- The current number of objects in the send queue to CarrierScan Server

The Antivirus Extension updates the file every two minutes.

APPENDIX A

Traffic Manager Statistics

This appendix describes the following statistics on the Traffic Manager **Monitor** tab:

My Proxy Statistics

The **My Proxy** statistics are divided into the following categories:

Summary

The following table describes the **Summary** statistics.

Statistic/Field	Description
Node	The name of the Traffic Edge node or cluster.
On/Off	Indicates if the Traffic Edge proxy is running (the proxy and manager services are running).
Objects Served	The total number of objects served by the Traffic Edge node since installation or since the statistics were cleared.
Ops/Sec	The number of operations per second processed by the Traffic Edge node.
Hit Rate	The percentage of requests served from the cache, averaged over the past 10 seconds.
Throughput (Mbit/sec)	The number of megabits per second passing through the Traffic Edge node (and cluster).

Statistic/Field	Description
HTTP Hit (ms)	The amount of time it takes for an HTTP object that is fresh in the cache to be served to the client.
HTTP Miss (ms)	The amount of time it takes for an HTTP object that is not in the cache or is stale to be served to the client.
More Detail	
cache hit rate	The percentage of HTTP requests served from the cache, averaged over the past 10 seconds, refreshed every 10 seconds.
errors	The percentage of HTTP requests that end in early hangups.
aborts	The percentage of aborted HTTP requests.
active clients	The current number of open client connections.
active servers	The current number of open origin server connections.
node IP address	The IP address assigned to the node; if virtual IP addressing is enabled, possibly several virtual IP addresses.
cache free space	The amount of free space in the cache.
HostDB hit rate	The ratio of host database hits to total host database lookups, averaged over a ten-second period.

Node

The following table describes the **Node** statistics.

Statistic	Description
Node Summary	
Status	Indicates if Traffic Edge is running on this node (active or inactive).
UP Since	The date and time Traffic Edge was started.
Clustering	Indicates if clustering is `On` or `Off` on this node.
Cache	
Document Hit Rate	The ratio of cache hits to total cache requests, averaged over 10 seconds, refreshed every 10 seconds.
Bandwidth Savings	The ratio of bytes served from the cache to total requested bytes, averaged over 10 seconds, refreshed every 10 seconds.
Cache Percent Free	The ratio of cache free space to total cache space.
In Progress	
Open Server Connections	The current number of open origin server connections.
Open Client Connections	The current number of open client connections.
Cache Transfers in Progress	The number of cache transfers (cache reads and writes) in progress.
Network	
Client Throughput (Mbit/Sec)	The number of megabits per second passing through the node (and cluster).
Transactions per Second	The number of transactions per second.

Statistic	Description
	Name Resolution
Host Database Hit Rate	The ratio of host database hits to total host database lookups, averaged over 10 seconds, refreshed every 10 seconds.
DNS Lookups per Second	The number of DNS lookups per second.

Graphs

Click the **Graphs** button to display the same statistics listed on the **Node** page (cache performance, current connections and transfers, network, and name resolution) but in graphical format. You can choose the statistics you want to present in a graph. For information about using the **Graphs** button, refer to *page 183*.

Alarms

Click the **Alarms** button to display the current Traffic Edge alarms. Traffic Edge signals an alarm when it detects a problem: for example, if the space allocated to event logs is full or if Traffic Edge cannot write to a configuration file.

For information about working with alarms, refer to *Working with Traffic Manager Alarms, on page 186*.

Protocol Statistics

The **Protocol** statistics are divided into the following categories:

- *HTTP*, described below
- *FTP*, described on *page 285*

HTTP

The following table describes the **HTTP** statistics.

Statistic	Description
	General
Client	
Total Document Bytes	The total amount of HTTP data served to clients since installation or since the statistics were cleared.
Total Header Bytes	The total amount of HTTP header data served to clients since installation or since the statistics were cleared.
Total Connections	The total number of HTTP client connections since installation or since the statistics were cleared.
Current Connections	The number of HTTP client connections currently open.
Transactions in Progress	The number of HTTP client transactions in progress.
Server	
Total Document Bytes	The total amount of HTTP data received from origin servers since installation or since the statistics were cleared.

Statistic	Description
Total Header Bytes	The total amount of HTTP header data received from origin servers since installation or since the statistics were cleared.
Total Connections	The total number of HTTP server connections since installation or since the statistics were cleared.
Current Connections	The current number of open HTTP server connections.
Transactions in Progress	The current number of HTTP server transactions.
Transaction	
Hits	
Fresh	The percentage of hits that are fresh and their average transaction times.
Stale Revalidated	The percentage of hits that are stale and revalidated and turn out to be still fresh and served, and their average transaction times.
Misses	
Now Cached	The percentage of requests for documents that were not in the cache (but are now) and their average transaction times.
Server No Cache	The percentage of requests for HTTP objects that were not in the cache but have server no-cache headers (cannot be cached) and their average transaction times.
Stale Reloaded	The percentage of misses that are revalidated and turn out to be changed, reloaded, and served, and their average transaction times.
Client No Cache	The percentage of requests for HTTP objects that were not in the cache but have client no-cache headers (cannot be cached) and their average transaction times.
Errors	
Connection Failures	The percentage of connect errors and their average transaction times.
Other Errors	The percentage of other errors and their average transaction times.
Aborted Transactions	
Client aborts	The percentage of client-aborted transactions and their average transaction times.
Questionable Client Aborts	The percentage of transactions that could possibly be client aborted and their average transaction times.
Partial Request Hangups	The percentage of early hangups (after partial requests) and their average transaction times.
Pre-Request Hangups	The percentage of pre-request hangups and their average transaction times.
Pre-Connect Hangups	The percentage of pre-connect hangups and their average transaction times.
Other Transactions	
Unclassified	The percentage of unclassified transactions and their average transaction times.

Statistic	Description
FTP over HTTP	
Connections	
Open Server Connections	The current number of open FTP server connections.
Successful PASV Connections	The number of successful PASV connections since installation or since the statistics were cleared.
Failed PASV Connections	The number of failed PASV connections since installation or since the statistics were cleared.
Successful PORT Connections	The number of successful PORT connections since installation or since the statistics were cleared.
Failed PORT Connections	The number of failed PORT connections since installation or since the statistics were cleared.
Cache Statistics	
Hits	The number of HTTP requests for FTP objects served from the cache since installation or since the statistics were cleared.
Misses	The number of HTTP requests for FTP objects forwarded directly to the origin server because the object is not in the cache or is stale, since installation or since the statistics were cleared.
Lookups	The number of times Traffic Edge looked up an HTTP request for an FTP object in the cache since installation or since the statistics were cleared.

FTP

The following table describes the **FTP** statistics.

Statistic	Description
Client	
Open Connections	The current number of open client connections.
Bytes Read	The number of client request bytes read since installation or since the statistics were cleared.
Bytes Written	The number of client request bytes written since installation or since the statistics were cleared.
Server	
Open Connections	The current number of open FTP server connections.
Bytes Read	The number of bytes read from FTP servers since installation or since the statistics were cleared.
Bytes Written	The number of bytes written to the cache since installation or since the statistics were cleared.
Operations	
File Hit	The number of FTP files served from the cache since installation or since the statistics were cleared.
File Misses	The number of FTP file misses since installation or since the statistics were cleared.
Change Directory Hits	The number of change directory hits since installation or since the statistics were cleared.

Change Directory Misses	The number of change directory misses since installation or since the statistics were cleared.
List Directory Hits	The number of list directory hits since installation or since the statistics were cleared.
List Directory Misses	The number of list directory misses since installation or since the statistics were cleared.

Streaming Media Statistics

The **Streaming Media** statistics are divided into the following categories:

- *QuickTime*, described below
- *Real Networks*, described on *page 287*
- *Windows Media*, described on *page 287*

QuickTime

The following table describes the **QuickTime** statistics.

Statistic	Description
	Live Streams
Current Live Streams	The current number of live streams that Traffic Edge is serving.
Number of Live Streams	The total number of live streams that Traffic Edge has served since installation or since the statistics were cleared.
Client	
Open Connections	The current number of open client connections.
Number of Requests	The number of QuickTime request connections Traffic Edge received from clients since installation or since the statistics were cleared.
Request Bytes	The total number of bytes that Traffic Edge has received from QuickTime clients since installation or since the statistics were cleared.
Response Bytes	The total number of bytes that Traffic Edge sent to QuickTime clients since installation or since the statistics were cleared.
Server	
Open Connections	The current number of open QuickTime server connections.
Number of Requests	The number of QuickTime request connections that Traffic Edge opened to QuickTime servers since installation or since the statistics were cleared.
Request Bytes	The total number of bytes Traffic Edge has sent to QuickTime servers since installation or since the statistics were cleared.
Response Bytes	The total number of bytes Traffic Edge has received from QuickTime servers since installation or since the statistics were cleared.

Real Networks

The following table describes the **Real Networks** statistics.

Statistic	Description
	Client
On Demand	
Open Connections	The current number of open client connections for on-demand streams.
Number of Requests	The number of Real media player requests Traffic Edge has received for on-demand streams since installation or since the statistics were cleared.
Response Bytes	The total number of bytes that Traffic Edge has sent to clients for on-demand streams since installation or since the statistics were cleared.
Live	
Open Connections	The current number of open client connections for live streams.
Number of Requests	The number of requests Traffic Edge has received from clients for live streams since installation or since the statistics were cleared.
	Server
On Demand	
Response Bytes	The total number of bytes Traffic Edge has received from origin RealServers for on-demand streams since installation or since the statistics were cleared.
	Cache
Total Bytes Hit	The number of bytes served from the cache since installation or since the statistics were cleared.
Total Bytes Missed	The number of bytes Traffic Edge has retrieved from the origin RealServer for on-demand requests since installation or since the statistics were cleared.

Windows Media

The following table describes the **Windows Media** statistics.

Statistic	Description
	Client
On Demand	
Number of Connections	The current number of open MMS-over-TCP, MMS-over-UDP, HTTP, and multicast connections for on-demand requests.
Number of Requests	The number of MMS-over-TCP, MMS-over-UDP, HTTP, and multicast requests received from clients for on-demand streams since installation or since the statistics were cleared.
Request Bytes	The total number of MMS-over-TCP, MMS-over-UDP, HTTP, and multicast request bytes received from clients for on-demand streams since installation or since the statistics were cleared.
Response Bytes	The total number of MMS-over-TCP, MMS-over-UDP, HTTP, and multicast response bytes sent to clients for on-demand streams since installation or since the statistics were cleared.

Live	
Number of Connections	The current number of open MMS-over-TCP, MMS-over-UDP, HTTP, and multicast connections for live streams.
Number of Requests	The number of MMS-over-TCP, MMS-over-UDP, HTTP, and multicast requests for live streams received from clients since installation or since the statistics were cleared.
Request Bytes	The total number of MMS-over-TCP, MMS-over-UDP, HTTP, and multicast request bytes received from clients for live streams since installation or since the statistics were cleared.
Response Bytes	The total number of MMS-over-TCP, MMS-over-UDP, HTTP, and multicast response bytes sent to clients for live streams since installation or since the statistics were cleared.
Server	
Open Connections	The current number of open origin server connections.
Number of Requests	The number of request connections that Traffic Edge has opened to origin servers since installation or since the statistics were cleared.
Request Bytes	The total number of bytes Traffic Edge has sent to origin servers since installation or since the statistics were cleared.
Response Bytes	The total number of bytes Traffic Edge has received from origin servers since installation or since the statistics were cleared.
Hit Rate	
Cumulative	The cumulative hit rate.
Instantaneous	The instantaneous hit rate.

Content Routing Statistics

The **Content Routing** statistics contain ICP information.

ICP Peering

The following table describes the **ICP Peering** statistics.

Statistic	Description
Queries Originating from This Node	
Query Requests	The number of HTTP requests that generated ICP query messages since installation or since the statistics were cleared.
Query Messages Sent	The total number of ICP query messages sent to ICP peers since installation or since the statistics were cleared (larger than the number of *Query Requests* if there are multiple ICP peers).
Peer Hit Messages Received	The number of ICP peer hit messages received in response to ICP queries from this node since installation or since the statistics were cleared.
Peer Miss Messages Received	The number of ICP peer miss messages received in response to ICP queries from this node since installation or since the statistics were cleared.

Total Responses Received	The number of response messages received from ICP peers (siblings and parents) since installation or since the statistics were cleared.
Average ICP Message Response Time (ms)	The average time for an ICP peer to respond to an ICP query message from this node. This is a cumulative average value.
Average ICP Request Time	The average time for an HTTP request (that is sent to ICP) to receive an ICP response. This is a cumulative average value.
Queries Originating from ICP Peers	
Query Messages Received	The number of ICP query messages received from remote ICP peers (siblings and parents) since installation or since the statistics were cleared.
Remote Query Hits	The number of successful cache lookups in response to queries from ICP peers since installation or since the statistics were cleared.
Remote Query Misses	The number of unsuccessful cache lookups in response to queries from ICP peers since installation or since the statistics were cleared.
Successful Response Messages Sent to Peers	The number of successful ICP messages written in response to ICP queries from remote ICP peers since installation or since the statistics were cleared.

Security Statistics

The **Security** statistics are divided into the following categories:

- *ARM Security*, described below
- *LDAP*, described on *page 290*
- *NTLM*, described on *page 290*
- *SOCKS*, described on *page 291*

ARM Security

The following table describes the **ARM Security** statistics.

Statistic	Description
Security Statistics	
TCP Dropped	The number of dropped TCP connections since Traffic Edge was started.
UDP Dropped	The number of dropped UDP connections since Traffic Edge was started.

LDAP

The following table describes the **LDAP** statistics.

Statistic	Description
Cache	
Hits	The number of LDAP hits in the Traffic Edge authentication cache since installation or since the statistics were cleared.
Misses	The number of LDAP misses in the Traffic Edge authentication cache since installation or since the statistics were cleared.
Errors	
Server	The number of LDAP server errors since installation or since the statistics were cleared.
Unsuccessful Authentication	
Authorization Denied	The number of times the LDAP Server denied authorization since installation or since the statistics were cleared.
Authorization Timeouts	The number of times authorization timed out since installation or since the statistics were cleared.
Authentication Cancelled	The number of times authentication was canceled since installation or since the statistics were cleared.

NTLM

The following table describes the **NTLM** statistics.

Statistic	Description
Cache	
Hits	The number of NTLM hits in the Traffic Edge authentication cache since installation or since the statistics were cleared.
Misses	The number of NTLM misses in the Traffic Edge authentication cache since installation or since the statistics were cleared.
Errors	
Server	The number of NTLM server errors since installation or since the statistics were cleared.
Unsuccessful Authentications	
Authorization Denied	The number of times the server denied authorization since installation or since the statistics were cleared.
Authentication Cancelled	The number of times authentication was canceled since installation or since the statistics were cleared.

SOCKS

The following table describes the **SOCKS** statistics.

Statistic	Description
Unsuccessful Connections	The number of unsuccessful connections to the SOCKS server since installation or since the statistics were cleared.
Successful Connections	The number of successful connections to the SOCKS server since installation or since the statistics were cleared.
Connections in Progress	The current number of open connections to the SOCKS server.

Subsystem Statistics

The **Subsystems** statistics are divided into the following categories:

- *Cache*, described below
- *Clustering*, described on *page 292*
- *Logging*, described on *page 292*

Cache

The following table describes the **Cache** statistics.

Statistic	Description
General	
Bytes Used	The number of bytes currently used by the cache.
Cache Size	The number of bytes allocated to the cache.
Ram Cache	
Bytes Used	The number of bytes currently used by the RAM cache.
Total Bytes Available	The total number of bytes available in the RAM cache.
Hits	The number of document hits from the RAM cache since Traffic Edge was started.
Misses	The number of document misses from the RAM cache since Traffic Edge was started (might be hits from the cache disk).
Reads	
In Progress	The number of cache reads in progress.
Hits	The number of cache reads completed since Traffic Edge was started.
Misses	The number of cache read misses since Traffic Edge was started.
Writes	
In Progress	The number of cache writes in progress.
Successes	The number of successful cache writes since Traffic Edge was started.
Failures	The number of failed cache writes since Traffic Edge was started.

Updates	
In Progress	The number of cache updates in progress. (An update occurs when Traffic Edge revalidates an object, finds it to be fresh, and updates the object header.)
Successes	The number of successful cache updates completed since Traffic Edge was started.
Failures	The number of cache update failures since Traffic Edge was started.
Removes	
In Progress	The number of document removes in progress. (A remove occurs when the Traffic Edge revalidates an object, finds it to be deleted on the origin server, and deletes it from the cache.)
Successes	The number of successful cache removes completed since Traffic Edge was started.
Failures	The number of cache remove failures since Traffic Edge was started.

Clustering

The following table describes the **Clustering** statistics.

Statistic	Description
Bytes Read	The number of bytes read by this node from other cluster nodes since Traffic Edge was started.
Bytes Written	The number of bytes this node has written to other cluster nodes since Traffic Edge was started.
Connections Open	The total number of intra-cluster connections opened since Traffic Edge was started.
Total Operations	The total number of cluster transactions since Traffic Edge was started.
Network Backups	The number of times this node encountered intra-cluster network congestion and reverted to proxy-only mode since Traffic Edge was started.
Clustering Nodes	The number of full-clustering nodes.

Logging

The following table describes the **Logging** statistics.

Statistic	Description
Currently Open Log Files	The number of event log files (formats) currently being written.
Space Used For Log Directory	The current amount of space being used by the logging directory, which contains all of the event and error logs.
Number of Access Events Logged	The number of access events written to log files since Traffic Edge installation or since the statistics were cleared. This counter represents one entry in one file. If multiple formats are being written, a single access will create multiple event log entries.

Number of Access Events Skipped	The number of access events skipped (because they were filtered out) since Traffic Edge installation or since the statistics were cleared.
Number of Error Events Logged	The number of access events that have been written to the event error log since installation or since the statistics were cleared.

Networking Statistics

The **Networking** statistics are divided into the following categories:

- *System*, described below
- *ARM*, described on *page 294*
- *WCCP*, described on *page 295*
- *DNS Proxy*, described on *page 296*
- *DNS Resolver*, described on *page 296*
- *Virtual IP*, described on *page 296*

System

The following table describes the **System** statistics.

Statistic/Field	Description
General	
Hostname	The hostname assigned to this Traffic Edge node.
Default Gateway	The IP address of the default gateway used to forward packets from this Traffic Edge node to other networks or subnets.
Search Domain	The search domain that this Traffic Edge node uses.
Primary DNS	The IP address of the primary DNS server that this Traffic Edge node uses to resolve hostnames.
Secondary DNS	The secondary DNS server that this Traffic Edge node uses to resolve hostnames.
Tertiary DNS	The third DNS server that this Traffic Edge node uses to resolve hostnames.
NIC (*interface_name*)	
Status	The status of the network interface.
Start on Boot	If the network interface enabled automatically when the Traffic Edge node booted.
IP address	The IP address assigned to the network interface.
Netmask	The netmask for the network interface.
Gateway	The gateway for the network interface.

ARM

The following table describes the **ARM** statistics.

Statistic	Description
Network Address Translation (NAT) Statistics	
Client Connections Natted	The number of client connections redirected transparently by the ARM since Traffic Edge was started.
Client Connections in Progress	The current number of open client connections with the ARM.
Total Packets Natted	The number of packets translated by the ARM since Traffic Edge was started.
DNS Packets Natted	The number of DNS packets translated by the ARM since Traffic Edge was started.
Bypass Statistics	
Total Connections Bypassed	The total number of connections bypassed by the ARM since Traffic Edge was started.
DNS Packets Bypassed	The number of DNS packets bypassed by the ARM since Traffic Edge was started.
HTTP Bypass Statistics	
Bypass on Bad Client Request	The number of requests forwarded directly to the origin server because Traffic Edge encountered nonHTTP traffic on port 80 since installation or since the statistics were cleared.
Bypass on 400	The number of requests forwarded directly to the origin server because an origin server returned a 400 error since installation or since the statistics were cleared.
Bypass on 401	The number of requests forwarded directly to the origin server because an origin server returned a 401 error since installation or since the statistics were cleared.
Bypass on 403	The number of requests forwarded directly to the origin server because an origin server returned a 403 error since installation or since the statistics were cleared.
Bypass on 405	The number of requests forwarded directly to the origin server because an origin server returned a 405 error since installation or since the statistics were cleared.
Bypass on 406	The number of requests forwarded directly to the origin server because an origin server returned a 406 error since installation or since the statistics were cleared.
Bypass on 408	The number of requests forwarded directly to the origin server because an origin server returned a 408 error since installation or since the statistics were cleared.
Bypass on 500	The number of requests forwarded directly to the origin server because an origin server returned a 500 error since installation or since the statistics were cleared.

WCCP

The following table describes the **WCCP** statistics.

Note WCCP 1.0 statistics display only if WCCP Version 1.0 is enabled. WCCP 2.0 statistics display only if WCCP Version 2.0 is enabled.

WCCPv1.0 Statistics	
Router Information	
Router IP address	The IP address of the router sending traffic to Traffic Edge.
Router Status	The status of the router: *up* (if Traffic Edge is able to communicate with the router) or *down* (if Traffic Edge is unable to communicate with the router).
Node Information	
My IP address	The IP address of this Traffic Edge node.
Percentage of Traffic Directed to This Node	The percentage of traffic directed to this Traffic Edge node.
Number of Heartbeats Received	The number of heartbeats received by this Traffic Edge node since Traffic Edge was started.
Protocol Information	
Leader's IP Address	The IP address of the leader in the WCCP cache farm.
Number of Active Nodes	The number of active Traffic Edge nodes in the WCCP cache farm.
WCCP v2.0 Statistics	
Group Name	The name of the Service Group: for example, HTTP.
Interface Address	The IP address of the interface on which the service group runs.
Group Leader	The IP address of the Traffic Edge node that is the leader for the service group.
Number of Caches	The number of caches participating in this service group.
Caches	The IP addresses of the caches participating in this service group.
Number of Routers	The number of routers participating in this service group.
Routers	The IP addresses of the routers participating in this service group.

DNS Proxy

The following table describes the **DNS Proxy** statistics.

Statistic	Description
DNS Proxy	
Total Requests	The total number of DNS requests received from clients since installation or since the statistics were cleared.
Hits	The number of DNS cache hits since installation or since the statistics were cleared.
Misses	The number of DNS cache misses since installation or since the statistics were cleared.

DNS Resolver

The following table describes the **DNS Resolver** statistics.

Statistic	Description
DNS Resolver	
Total Lookups	The total number of DNS lookups (queries to name servers) since installation or since the statistics were cleared.
Successes	The total number of successful DNS lookups since installation or since the statistics were cleared.
Average Lookup Time (ms)	The average DNS lookup time.
Host DataBase	
Total Lookups	The total number of lookups in the Traffic Edge host database since installation or since the statistics were cleared.
Total Hits	The total number of host database lookup hits since installation or since the statistics were cleared.
Average TTL (min)	The average time to live in minutes.

Virtual IP

The **Virtual IP** table displays the virtual IP addresses that are managed by the Traffic Edges in the cluster.

MRTG Statistics

MRTG (Multi Router Traffic Grapher) is a graphing tool that enables you to monitor Traffic Edge performance and analyze network traffic. MRTG provides a variety of graphs that show information about virtual memory usage, client connections, document hit rates, hit and miss rates, and so on. MRTG uses five-minute intervals to formulate the statistics and provides useful historical information. MRTG provides the following categories of information.

Statistics	Description
Overview	Displays a subset of the graphs available in MRTG.
Daily	Displays graphs that provide historical information for the current day.
Weekly	Displays graphs that provide historical information for the current week.
Monthly	Displays graphs that provide historical information for the current month.
Yearly	Displays graphs that provide historical information for the current year.

IMPORTANT To run MRTG in UNIX, you must have Perl Version 5.005 or later installed on your Traffic Edge system. To run MRTG in Windows, you must have Windows Services for UNIX (SFU) 2.0 or later installed on your Traffic Edge system.

APPENDIX B

Traffic Manager Configuration Options

This appendix describes the following configuration options on the Traffic Manager **Configure** tab:

My Proxy

My Proxy configuration options are divided into the following categories:

- *Basic*, described below
- *UI Setup*, described on *page 302*
- *Snapshots*, described on *page 304*
- *Logs*, described on *page 305*

Basic

The following table describes the **Basic** proxy configuration options.

Option	Description
	General
Restart	Restarts the Traffic Edge proxy and manager services (the `traffic_server` and `traffic_manager` processes). You must restart the Traffic Edge proxy and manager services after modifying certain configuration options. In a cluster configuration, the **Restart** button restarts the Traffic Edge proxy and manager services on all the nodes in the cluster.
Clear Statistics	Resets the statistics on the **Monitor** tab to zero on this node.

Proxy Name	Specifies the name of your Traffic Edge node (by default, this is the hostname of the machine running Traffic Edge). If this node is part of a cluster, this option specifies the name of the Traffic Edge cluster (in a Traffic Edge cluster, all nodes must share the same name).
Alarm E-Mail	Specifies the email address to which Traffic Edge sends alarm notifications.
Features	
General: SNMP	Enables or disables the Traffic Edge SNMP agent. The Traffic Edge SNMP agent supports access to two management information bases (MIBs): MIB-2 (a standard MIB) and the Inktomi Traffic Edge MIB. Descriptions of the Traffic Edge MIB variables are provided in the `inktomi-ts-mib.my` file in the Traffic Edge `config/mibs` directory. The Traffic Edge MIB contains both node-specific and cluster-wide information. You should configure your system so that only certain hosts can access these MIBs. You configure access control and SNMP trap destinations in the `snmpd.cnf` file in the Traffic Edge `config` directory; refer to *snmpd.cnf, on page 449*.
Protocols: NNTP	Not supported for this release.
Protocols: FTP	Enables or disables processing of FTP requests from FTP clients. When enabled, Traffic Edge accepts FTP requests from FTP clients. When disabled, Traffic Edge does not accept FTP requests from FTP clients. If you change this option, you must restart Traffic Edge.
Streaming Media: QuickTime	Enables or disables processing of QuickTime requests.
Streaming Media: Real Networks	Enables or disables processing of Real Networks requests.
Streaming Media: Windows Media	Enables or disables processing of Windows Media requests.
Security: LDAP	Enables or disables LDAP proxy authentication. When enabled, you can ensure that users are authenticated by an LDAP server before accessing content from the Traffic Edge cache. Refer to *Using LDAP Proxy Authentication, on page 215*. If you change this option, you must restart Traffic Edge.
Security: Radius	Enables or disables RADIUS proxy authentication. When enabled, you can ensure that users are authenticated by a RADIUS server before accessing content from the Traffic Edge cache. Refer to *Using RADIUS Proxy Authentication, on page 218*. If you change this option, you must restart Traffic Edge.
Security: NTLM	Enables or disables NTLM proxy authentication. When enabled, you can ensure that users in a Windows network are authenticated by a Domain Controller before accessing content from the Traffic Edge cache. Refer to *Using NTLM Proxy Authentication, on page 221*. If you change this option, you must restart Traffic Edge.

Security: SSL Termination	Enables or disables the SSL termination option so that you can secure connections in reverse proxy mode between a client and a Traffic Edge and/or between Traffic Edge and an origin server. Refer to *Using SSL Termination, on page 226*. If you change this option, you must restart Traffic Edge.
Security: SOCKS	Enables or disables the SOCKS option. When enabled, Traffic Edge can talk to your SOCKS servers. Refer to *Configuring SOCKS Firewall Integration, on page 210*. If you change this option, you must restart Traffic Edge.
Networking: ARM	Enables or disables the Traffic Edge ARM, which is used for transparent proxy caching, IP spoofing, and ARM security. Refer to *Chapter 6, Transparent Proxy Caching*. If you change this option, you must restart Traffic Edge.
Networking: WCCP	Enables or disables WCCP. Enable this option if you are using a WCCP-enabled router for transparent proxy caching. Refer to *Using a WCCP-Enabled Router, on page 110*. If you change this option, you must restart Traffic Edge.
Networking: DNS Proxy	Enables or disables the DNS proxy caching option. When enabled, Traffic Edge can resolve DNS requests on behalf of clients. This option offloads remote DNS servers and reduces response time for DNS lookups. Refer to *Chapter 11, DNS Proxy Caching*. If you change this option, you must restart Traffic Edge.
Networking: Virtual IP	Enables or disables the virtual IP failover option. When enabled, Traffic Edge maintains a pool of virtual IP addresses that it assigns to the nodes in a cluster as necessary. Refer to *Using Virtual IP Failover, on page 155*.
Clustering	
Cluster: Type	Specifies the Traffic Edge clustering mode: Select **Single Node** to run this Traffic Edge as a single node. This Traffic Edge node will not be part of a cluster. Select **Management Clustering** to run management-only clustering mode. The Traffic Edge nodes in the cluster share configuration information and you can administer all the nodes at the same time. Select **Full Cache Clustering** to run full-clustering mode. In full-clustering mode, as well as sharing configuration information, a Traffic Edge cluster distributes its cache across its nodes into a single, virtual object store, rather than replicating the cache node by node. For more information about clustering mode, refer to *Understanding Traffic Edge Clusters, on page 151*. If you change this option, you must restart Traffic Edge.
Clear Statistics	Resets to zero the statistics on the **Monitor** tab that are calculated since Traffic Edge was started or installed on all nodes in the cluster. This button displays only when **Management Clustering** or **Full Cache Clustering** is enabled.

UI Setup

The following table describes the **UI Setup** configuration options.

Option	Description
	General
UI Port	Specifies the port on which browsers can connect to Traffic Manager. The port must be on Traffic Edge and it must be dedicated to Traffic Edge use. The default port is 8081. If you change this option, you must restart Traffic Edge.
HTTPS: Enable/Disable	Enables/disables support for SSL connections to Traffic Manager. SSL provides protection for remote administrative monitoring and configuration. To use SSL for Traffic Manager connections, you must install an SSL certificate on the Traffic Edge node. For more information, refer to *Using SSL for Secure Administration, on page 208*.
HTTPS: Certificate File	Specifies the name of the SSL certificate file used to authenticate users that want to access Traffic Manager. If you change this option, you must restart Traffic Edge.
Monitor Refresh Rate	Specifies how often Traffic Manager refreshes the statistics on the **Monitor** tab. The default value is 30 seconds.
	Login
Basic Authentication	Enables or disables basic authentication. When enabled, Traffic Edge checks the administrator login and password or the username and password (if user accounts have been configured) whenever a user tries to access Traffic Manager.
Administrator: Login	Specifies the administrator login. The administrator login is the master login that has access to both Configure and Monitor mode in Traffic Manager. Traffic Edge checks the administrator login only if the **Basic Authentication** option is enabled.
Administrator: Password	Lets you change the administrator password that controls access to Traffic Manager. To change the password, enter the current password in the **Old Password** field and then enter the new password in the **New Password** field. Retype the new password in the **New Password (Retype)** field and then click the **Apply** button. Note: Traffic Edge checks the administrator login and password only if the **Basic Authentication** option is enabled. During installation, you select the administrator password. The installer automatically encrypts the password and stores the encryptions in the `records.config` file so that no one can read them. Each time you change the password in Traffic Manager, Traffic Edge updates the `records.config` file. If you forget the administrator password and cannot access Traffic Manager, refer to *How do you access Traffic Manager if you forget the master administrator password?, on page 475*.

Additional Users	Lists the current user accounts and lets you add new user accounts. User accounts determine who can access Traffic Manager and which activities they can perform. You can create a list of user accounts if a single administrator login and password is not sufficient security for your needs. To create a new account, enter the user login in the **New User** field and then enter the user password in the **New Password** field. Retype the user password in the **New Password (Retype)** field and then click the **Apply** button. The new user displays in the table. From the **Access** drop-down list in the table, select the activities that the new user can perform (No Access, Monitor Only, Monitor and View Configuration, or Monitor and Modify Configuration). For more information about user accounts, refer to *Creating a List of User Accounts, on page 207*. Traffic Edge checks the user login and password only if the **Basic Authentication** option is enabled.
Access	
Access Control	Displays a table listing the rules in the `mgmt_allow.config` file that specify the remote hosts allowed to access Traffic Manager. The entries in this file ensure that only authenticated users can change Traffic Edge configuration options and view performance and network traffic statistics. By default, all remote hosts are allowed to access Traffic Manager.
Refresh	Updates the table to display the most up-to-date rules in the `mgmt_allow.config` file. Click this button after you have added or modified rules with the configuration file editor.
Edit File	Opens the configuration file editor so that you can edit and add rules to the `mgmt_allow.config` file. The configuration file editor page is described below.
mgmt_allow.config Configuration File Editor	
rule display box	Lists the `mgmt_allow.config` file rules. You must select a rule to edit it. The buttons on the left of the box allow you to delete or move the selected rule up or down in the list. Traffic Edge applies the rules in the order listed, starting from the top.
Add	Adds a new rule to the rule display box at the top of the configuration file editor page. You must enter information in the fields provided before you click this button.
Set	Updates the rule display box at the top of the configuration file editor page with any configuration changes you make**.** You must select a rule and change its properties before you click this button.
IP Action	Lists the type of rules you can add. An **ip_allow** rule allows the remote hosts specified in the **Source IP** field to access Traffic Manager. An **ip_deny** rule denies the remote hosts specified in the **Source IP** field access to Traffic Manager.
Source IP	Specifies the IP addresses that are allowed or denied access to Traffic Manager. You can enter either a single IP address (111.111.11.1) or a range of IP addresses (0.0.0.0-255.255.255.255).
Clear Fields	Clears all the fields provided.

Apply	Applies the configuration changes. If you set invalid rules, Traffic Edge displays a warning message when you click the **Apply** button.
Close	Exits the configuration file editor. Important: You must click the **Apply** button before you click **Close**; otherwise, all configuration changes will be lost.

Snapshots

The following table describes the **Snapshot** configuration options

The **FTP Server** and the **Floppy Disk** tabs display only on a Linux system.

File System	
Change Snapshot Directory	Specifies the directory in which snapshots are stored on this Traffic Edge node.
Snapshots: Save Snapshot	Specifies the name of the configuration snapshot you want to take. Click the **Apply** button to save the configuration on the local node. Traffic Edge saves the configuration snapshot in the directory specified in the **Change Snapshot Directory** field, described above. Inktomi recommends that you take a snapshot before performing system maintenance or attempting to tune system performance. Taking a snapshot only takes a few seconds and it can save you hours of correcting configuration mistakes.
Snapshots: Restore/Delete Snapshot	Lists the snapshots that are stored on this Traffic Edge node. Select the snapshot that you want to restore or delete from the drop-down list.
Snapshots: Restore Snapshot from "*directory_name*" Directory	Restores the snapshot selected in the **Restore/ Delete Snapshot** drop-down box. In a cluster configuration, this button restores the snapshot on all nodes in the cluster.
Snapshots: Delete Snapshot from "*directory_name*" Directory	Deletes the snapshot selected in the **Restore/ Delete Snapshot** drop-down box.
FTP Server	
Login Information: FTP server	Specifies the name of the FTP server from which you want to restore a configuration snapshot or to which you want to save a configuration snapshot.
Login Information: Login	Specifies the login needed to access the FTP Server.
Login Information: Password	Specifies the password needed to access the FTP Server.
Login Information: Remote Directory	Specifies the directory on the FTP server from which you want restore, or in which you want to save a configuration snapshot.
Restore Snapshot	Lists the configuration snapshots on the FTP server that you can restore. This field appears after you have logged on to the FTP server successfully.

Save Snapshot to FTP Server	Specifies the name of the configuration snapshot you want to take and save on the FTP server. This field appears after you have logged on to the FTP server successfully.
Floppy Disk	
Select Floppy Drive	Specifies the floppy disk drives available from which you want to restore or to which you want to save a configuration snapshot.
Restore Snapshot	Lists the configuration snapshots on the floppy disk that you can restore. This field appears after you have selected the floppy drive from the **Select Floppy Drive** field.
Save Snapshot to Floppy Disk	Specifies the name of the configuration snapshot you want to take and save on the floppy disk. This field appears after you have selected the floppy drive from the **Select Floppy Drive** field.

Logs

The following table describes the **Logs** configuration options.

System	
Log File	Lists the system log files you can view, delete, or copy to your local system. Traffic Edge lists the system log files logged with the system-wide logging facility `syslog` under the daemon facility.
Action: Display the selected log file	When enabled, Traffic Edge displays the entire system log file selected in the **Log File** drop-down list.
Action: Display last lines of the selected file	When enabled, Traffic Edge displays the last specified number of lines in the selected system log file.
Action: Display lines that match in the selected log file	When enabled, Traffic Edge displays all the lines in the selected system log file that match the specified string.
Remove the selected log file	When enabled, Traffic Edge deletes the selected log file.
Action: Save the selected log file in local filesystem	When enabled, Traffic Edge saves the selected log file on the local system in a location you specify.
Access	
Log File	Lists the event or error log files you can view, delete or copy to your local system. Traffic Edge lists the event log files located in the directory specified in the **Logging Directory** field under **Subsystems**/**Logging** and by the configuration variable `proxy.config.log2.logfile_dir` in the `records.config` file. The default directory is `logs` in the Traffic Edge installation directory.
Action: Display the selected log file	When enabled, Traffic Edge displays the entire event or error log file selected in the **Log File** drop-down list.
Action: Display last lines of the selected file	When enabled, Traffic Edge displays the last specified number of lines in the event or error log file selected from the **Log File** drop-down list.

Action: Display lines that match in the selected log file	When enabled, Traffic Edge displays all the lines in the selected event or error log file that match the specified string.
Remove the selected log file	When enabled, Traffic Edge deletes the selected log file.
Action: Save the selected log file in local filesystem	When enabled, Traffic Edge saves the selected log file on the local system in a location you specify.

Protocols

The **Protocol** configuration options are divided into the following categories:

- *HTTP*, described below
- *HTTP Responses*, described on *page 314*
- *HTTP Scheduled Update*, described on *page 315*
- *FTP*, described on *page 316*

HTTP

The following table describes the **HTTP** configuration options.

Option	Description
	General
HTTP Proxy Server Port	Specifies the port that Traffic Edge uses when acting as a web proxy server for HTTP traffic or when serving HTTP requests transparently. The default port is 8080. If you change this option, you must restart Traffic Edge.
URL Expandomatic	Enables or disables `.com` domain expansion. When enabled, Traffic Edge attempts to resolve unqualified hostnames by redirecting them to the expanded address, prepended with `www.` and appended with `.com.`; for example, if a client makes a request to `company`, Traffic Edge redirects the request to `www.company.com`. Note: If local domain expansion is enabled (refer to *Local Domain Expansion, on page 347*), Traffic Edge attempts local domain expansion before `.com` domain expansion; Traffic Edge tries `.com` domain expansion only if local domain expansion fails.

PUSH Method	Enables or disables the HTTP PUSH option that lets you to deliver HTTP content directly to the cache without user request. Important: If you enable this option, you must also specify a filtering rule in the `filter.config` file to allow only certain machines to push content into the cache. Refer to *Pushing Content into the Cache, on page 44*.
HTTPS Redirect	Specifies the range of ports used for tunneling. Traffic Edge allows tunnels only to the specified ports; for example, to retrieve an object using HTTPS via Traffic Edge, a tunnel must be established via Traffic Edge to an origin server.
FTP over HTTP: Anonymous Password	Specifies the anonymous password Traffic Edge must use for FTP server connections that require a password. This option affects FTP requests from HTTP clients.
FTP over HTTP: Data Connection Mode	An FTP transfer requires two connections: a control connection to inform the FTP server of a request for data and a data connection to send the data. Traffic Edge always initiates the control connection. FTP mode determines whether Traffic Edge or the FTP server initiates the data connection. Select **PASV then PORT** for Traffic Edge to attempt PASV connection mode first. If PASV mode fails, Traffic Edge tries PORT mode and initiates the data connection. If successful, the FTP server accepts the data connection. Select **PASV only** for Traffic Edge to initiate the data connection to the FTP server. This mode is firewall friendly, but some FTP servers do not support it. Select **PORT only** for the FTP server to initiate the data connection and for Traffic Edge to accept the connection. The default value is **PASV then PORT**.
Cacheability	
Caching: HTTP Caching	Enables or disables HTTP caching. When enabled, Traffic Edge serves HTTP requests from the cache. When disabled, Traffic Edge acts as a proxy server and forwards all HTTP requests directly to the origin server.
Caching: FTP over HTTP Caching	Enables or disables FTP-over-HTTP caching. When enabled, Traffic Edge serves FTP requests from HTTP clients from the cache. When disabled, Traffic Edge acts as a proxy server and forwards all FTP requests from HTTP clients directly to the FTP server.

Behavior: Required Headers	Specifies the minimum header information required for an HTTP object to be cacheable. Select **An Explicit Lifetime Header** to cache only HTTP objects with Expires or max-age headers. Select **A Last-Modified Header** to cache only HTTP objects with last-modified headers. Select **No Required Headers** to cache HTTP objects that do not have Expires, max-age, or last-modified headers. This is the default option. **Caution**: By default, Traffic Edge caches all objects (including objects with no headers). Inktomi recommends that you change the default setting only for specialized proxy situations. If you configure Traffic Edge to cache only HTTP objects with Expires or max-age headers, the cache hit rate will be seriously reduced (very few objects have explicit expiration information).
Behavior: When to Revalidate	Specifies how Traffic Edge evaluates HTTP object freshness in the cache: Select **Never Revalidate** to never revalidate HTTP objects in the cache with the origin server (Traffic Edge considers all HTTP objects in the cache to be fresh). Select **Always Revalidate** to always revalidate HTTP objects in the cache with the origin server (Traffic Edge considers all HTTP objects in the cache to be stale). Select **Revalidate if Heuristic Expiration** to verify the freshness of an HTTP object with the origin server if the object contains no Expires or Cache-control headers; Traffic Edge considers all HTTP objects without Expires or Cache-control headers to be stale. Select **Use Cache Directive or Heuristic** to verify the freshness of an HTTP object with the origin server when Traffic Edge considers the object in the cache to be stale according to object headers, absolute freshness limit, and/or rules in the `cache.config` file. This is the default option. For more information about revalidation, refer to *Revalidating HTTP Objects, on page 37*.

Behavior: Add “no-cache” to MSIE Requests	Specifies when Traffic Edge adds `no-cache` headers to requests from Microsoft Internet Explorer. Certain versions of Microsoft Internet Explorer do not request cache reloads from reverse proxies and transparent caches when the user presses the browser **Refresh** button. This can prevent content from being loaded directly from the origin servers. You can configure Traffic Edge to treat Microsoft Internet Explorer requests more conservatively, providing fresher content at the cost of serving fewer documents from cache. Select **To All MSIE Requests** to always add `no-cache` headers to all requests from Microsoft Internet Explorer. Select **To IMS MSIE Requests** to add `no-cache` headers to IMS (If Modified Since) Microsoft Internet Explorer requests. Select **Not to Any MSIE Requests** to never add `no-cache` headers to requests from Microsoft Internet Explorer.
Behavior: Ignore “no-cache” in Client Requests	When enabled, Traffic Edge ignores `no-cache` headers in client requests and serves the requests from the cache. When disabled, Traffic Edge does not serve requests with `no-cache` headers from the cache but forwards them to the origin server.
Freshness: Minimum Heuristic Lifetime	Specifies the minimum amount of time that an HTTP object can be considered fresh in the cache.
Freshness: Maximum Heuristic Lifetime	Specifies the maximum amount of time that an HTTP object can be considered fresh in the cache.
Freshness: FTP Document Lifetime	Specifies the maximum amount of time that an FTP file can stay in the Traffic Edge cache. This option affects FTP requests from HTTP clients only.
Maximum Alternates	Specifies the maximum number of alternate versions of HTTP objects Traffic Edge can cache. **Caution**: If you enter `0` (zero), there is no limit to the number of alternates cached. If a popular URL has thousands of alternates, you might observe increased cache hit latencies (transaction times) as Traffic Edge searches over the thousands of alternates for each request. In particular, some URLs can have large numbers of alternates due to cookies. If Traffic Edge is set to vary on cookies, you might encounter this problem.
Vary Based on Content Type: Enable/Disable	Enables or disables caching of alternate versions of HTTP documents that do not contain the Vary header. If no Vary header is present, Traffic Edge will vary on the headers specified below, depending on the document's content type.
Vary by Default on Text	Specifies the header field on which Traffic Edge varies for text documents.
Vary by Default on Images	Specifies the header field on which Traffic Edge varies for images.

Vary by Default on Other Document Types	Specifies the header field on which Traffic Edge varies for anything other than text and images.
Dynamic Caching: Caching Documents with Dynamic URLs	When enabled, Traffic Edge attempts to cache dynamic content. Content is considered dynamic if it contains a question mark (?), a semicolon (;), cgi, or if it ends in .asp. **Caution**: Inktomi recommends that you configure Traffic Edge to cache dynamic content for specialized proxy situations only.
Dynamic Caching: Caching Response to Cookies	Specifies how responses to requests that contain cookies are cached: Select **Cache All but Text** to cache cookies that contain any type of content except text. This is the default option. Select **Cache Only Image Types** to cache cookies that contain images only. Select **Cache Any Content-Type** to cache cookies that contain any type of content. Select **No Cache on Cookies** to *not* cache cookies at all.
Caching Policy/Forcing Document Caching	Displays a table listing the rules in the `cache.config` file that specify how a particular group of URLs should be cached. This file also lets you force caching of certain URLs for a specific amount of time.
Refresh	Updates the table to display the most up-to-date rules in the `cache.config` file. Click this button after you have added or modified rules with the configuration file editor.
Edit File	Opens the configuration file editor so that you can edit and add rules to the `cache.config` file. The configuration file editor page is described below.
cache.config Configuration File Editor	
Rule display box	Lists the `cache.config` file rules. You must select a rule to edit it. The buttons on the left of the box allow you to delete or move the selected rule up or down in the list.
Add	Adds a new rule to the rule display box at the top of the configuration file editor page. You must enter information in the fields provided before you click this button.
Set	Updates the rule display box at the top of the configuration file editor page with any configuration changes you make**.** You must select a rule and change its properties before you click this button.

Rule Type	Lists the type of rules you can add to the `cache.config` file: A **never-cache** rule configures Traffic Edge to never cache specified objects. An **ignore-no-cache** rule configures Traffic Edge to ignore all Cache-Control: no-cache headers. An **ignore-client-no-cache** rule configures Traffic Edge to ignore Cache-Control: no-cache headers from client requests. An **ignore-server-no-cache** rule configures Traffic Edge to ignore Cache-Control: no-cache headers from origin server responses. A **pin-in-cache** rule configures Traffic Edge to keep objects in the cache for a specified time. A **revalidate** rule configures Traffic Edge to consider objects fresh in the cache for a specified time. A **ttl-in-cache** rule configures Traffic Edge to serve certain HTTP objects from the cache for the amount of time specified in the **Time Period** field regardless of certain caching directives in the HTTP request and response headers.
Primary Destination Type	Lists the primary destination types: **dest_domain** is a requested domain name. **dest_host** is a requested hostname. **dest_ip** is a requested IP address. **url_regex** is a regular expression to be found in a URL.
Primary Destination Value	Specifies the value of the primary destination type; for example, if the **Primary Destination Type** is **dest_ip**, the value for this field can be 123.456.78.9.
Additional Specifiers: Time Period	Specifies the amount of time that applies to the **revalidate**, **pin-in-cache**, and **ttl-in-cache** rule types. The following time formats are allowed: `d` for days: for example 2d `h` for hours: for example, 10h `m` for minutes: for example, 5m `s` for seconds: ffor example, 20s mixed units: for example, 1h15m20s
Secondary Specifiers: Time	Specifies a time range, such as 08:00-14:00.
Secondary Specifiers: Prefix	Specifies a prefix in the path part of a URL.
Secondary Specifiers: Suffix	Specifies a file suffix in the URL.
Secondary Specifiers: Source IP	Specifies the IP address of the client.
Secondary Specifiers: Port	Specifies the port in a requested URL. You can specify a single port or a range of ports.
Secondary Specifiers: Method	Specifies a request URL method.
Secondary Specifiers: Scheme	Specifies the protocol of a requested URL.
Secondary Specifiers: MIXT Scheme	Specifies the media protocol type of a requested URL.

Clear Fields	Clears all the fields provided.
Apply	Applies the configuration changes. If you set invalid rules, Traffic Edge displays a warning message when you click the **Apply** button.
Close	Exits the configuration file editor. Important: You must click the **Apply** button before you click **Close**; otherwise, all configuration changes will be lost.
	Privacy
Insert Headers: Client-IP	When enabled, Traffic Edge inserts the `Client-IP` header in outgoing requests if the request does not contain an `Client-IP` header.
Remove Headers: Client-IP	When enabled, Traffic Edge removes the `Client-IP` header from outgoing requests to protect the privacy of your users.
Remove Headers: Cookie	When enabled, Traffic Edge removes the `Cookie` header from outgoing requests to protect the privacy of your users. The `Cookie` header often identifies the user that makes a request.
Remove Headers: From	When enabled, Traffic Edge removes the `From` header from outgoing requests to protect the privacy of your users. The `From` header identifies the client's email address.
Remove Headers: Referer	When enabled, Traffic Edge removes the `Referer` header from outgoing requests to protect the privacy of your users. The `Referer` header identifies the web link that the client selects.
Remove Headers: User-Agent	When enabled, Traffic Edge removes the `User-Agent` header from outgoing requests to protect the privacy of your users. The `User-Agent` header identifies the agent that is making the request, usually a browser.
Remove Headers: Remove Others	Specifies headers other than `From`, `Referer`, `User-Agent`, and `Cookie`, that you want to remove from outgoing requests to protect the privacy of your users.
	Timeouts
Keep-Alive Timeouts: Client	Specifies how long the Traffic Edge should keep connections to clients open for a subsequent request after a transaction ends. Each time Traffic Edge opens a connection to accept a client request, it handles the request and then keeps the connection alive for the timeout period you specify. If the client does not make another request before the timeout expires, Traffic Edge closes the connection; otherwise, the timeout period starts again. The client can close the connection at any time.

Keep-Alive Timeouts: Origin Server	Specifies how long the Traffic Edge should keep connections to origin servers open for a subsequent transfer of data after a transaction ends. Each time Traffic Edge opens a connection to download data from an origin server, it downloads the data and then keeps the connection alive for the timeout period you specify. If Traffic Edge does not need to make a subsequent request for data before the timeout expires, it closes the connection. If it does, the timeout period starts again. The origin server can close the connection at any time.
Inactivity Timeouts: Client	Specifies how long the Traffic Edge should keep connections to clients open if a transaction stalls. If Traffic Edge stops receiving data from a client or the client stops reading the data, the Traffic Edge closes the connection when this timeout expires. The client can close the connection at any time.
Inactivity Timeouts: Origin Server	Specifies how long the Traffic Edge should keep connections to origin servers open if the transaction stalls. If Traffic Edge stops receiving data from an origin server, it will not close the connection until this timeout has expired. The origin server can close the connection at any time.
Active Timeouts: Client	Specifies how long Traffic Edge can remain connected to a client. If the client does not finish making a request (reading and writing data) before this timeout expires, Traffic Edge closes the connection. The client can close the connection at any time. The default value of 0 (zero) specifies that there is no timeout.
Active Timeouts: Origin Server	Specifies how long Traffic Edge can wait for fulfillment of a connection request to an origin server. If Traffic Edge does not establish connection to an origin server before this timeout expires, Traffic Edge terminates the connection request. The origin server can close the connection at any time. The default value of 0 (zero) specifies that there is no timeout.
FTP Control Connection Timeout	Specifies how long Traffic Edge can wait for a response from an FTP server. If the FTP server does not respond within the amount of time you specify, Traffic Edge abandons the client request for data. This option affects FTP requests from HTTP clients only.

HTTP Responses

The following table describes the **HTTP Response** configuration options.

Option	Description
	General
Response Suppression Mode	If Traffic Edge detects an HTTP problem with a particular client transaction (such as unavailable origin servers, authentication requirements, and protocol errors), it sends an HTML response to the client browser. Traffic Edge has a set of hard-coded default response pages that explain each HTTP error in detail to the client. Select **Always Suppressed** if you do not want to send HTTP responses to clients. Select **Intercepted Traffic Only** if you want to send HTTP responses to nontransparent traffic only. (This option is useful when Traffic Edge is running transparently and you do not want to indicate the presence of a cache.) Select **Never Suppressed** if you want to send HTTP responses to all clients. If you change this option, you must restart Traffic Edge.
	Custom
Custom Responses	You can customize the responses Traffic Edge sends to clients. By default, the responses you can customize are located in the Traffic Edge `config/body_factory/default` directory. Select **Enabled Language-Targeted Response** to send your custom responses to clients in the language specified in the `Accept-Language` header. Select **Enabled in "default" Directory Only** to send the custom responses located in the `default` directory to clients. Select **Disabled** to disable the custom responses. If **Never Suppressed** or **Intercepted Traffic Only** is selected for the **Response Suppression Mode** option, Traffic Edge sends the hard-coded default responses. If you change this option, you must restart Traffic Edge.
Custom Response Logging	When enabled, Traffic Edge sends a message to the error log each time custom responses are used or modified. If you change this option, you must restart Traffic Edge.
Custom Response Template Directory	Specifies the directory where the custom responses are located. The default location is the Traffic Edge `config/body_factory` directory. If you change this option, you must restart Traffic Edge.

HTTP Scheduled Update

The following table describes the **HTTP Scheduled Update** configuration options.

Option	Description
General	
Scheduled Update	Enables or disables the scheduled update option. When this option is enabled, Traffic Edge can automatically update certain objects in the local cache at a specified time.
Maximum Concurrent Updates	Specifies the maximum number of simultaneous update requests allowed at any point in time. This option enables you to prevent the scheduled update process from overburdening the host. The default value is 100.
Retry on Update Error: Count	Specifies the number of times Traffic Edge can retry the scheduled update of a URL in the event of failure. The default value is 10 times.
Retry on Update Error: Interval	Specifies the delay in seconds between each scheduled update retry for a URL in the event of failure. The default value is 2 seconds.
Update URLs	
Force Immediate Update	When enabled, Traffic Edge overrides the scheduling expiration time for all scheduled update entries and initiates updates every 10 seconds.
Scheduled Object Update	Displays a table listing the rules in the `update.config` file that control how Traffic Edge performs a scheduled update of specific local cache content.
Refresh	Updates the table to display the most up-to-date rules in the `update.config` file. Click this button after you have added or modified rules with the configuration file editor.
Edit File	Opens the configuration file editor so that you can edit and add rules to the `update.config` file. The configuration file editor page is described below.
update.config Configuration File Editor	
rule display box	Lists the `update.config` file rules. You must select a rule to edit it. The buttons on the left of the box allow you to delete or move the selected rule up or down in the list.
Add	Adds a new rule to the rule display box at the top of the configuration file editor page. You must enter information in the fields provided before you click this button.
Set	Updates the rule display box at the top of the configuration file editor page with any configuration changes you make**.** You must select a rule and change its properties before you click this button.
URL	Specifies the URL to be updated.
Request Headers (optional)	Specifies the list of headers (separated by semi-colons) passed in each `GET` request. You can define any request header that conforms to the HTTP specification. The default is no request header.
Offset Hour	Specifies the base hour used to derive the update periods. The range is 00-23 hours.

Interval	The interval, in seconds, at which updates should occur, starting at Offset hour.
Recursion Depth	The depth to which referenced URLs are recursively updated, starting at the given URL; for example, a recursion depth of 1 will update the given URL, as well as all URLs immediately referenced by links from the original URL.
Clear Fields	Clears all the fields provided.
Apply	Applies the configuration changes. If you set invalid rules, Traffic Edge displays a warning message when you click the **Apply** button.
Close	Exits the configuration file editor. Important: You must click the **Apply** button before you click **Close**; otherwise, all configuration changes will be lost.

FTP

The following table describes the **FTP** configuration options.

The **FTP** configuration options display on the **Configure** tab only if you have enabled **FTP.**

Option	Description
	General
FTP Proxy Server Port	Specifies the port that Traffic Edge uses to accept FTP requests. The default port is 21.
Listening Port Configuration	Specifies how FTP opens a listening port for a data transfer. Select **Default Settings** to let the operating system choose an available port. Traffic Edge sends 0 and retrieves the new port number if the listen succeeds. Select **Specify Range** if you want the listening port to be determined by the range of ports specified in the **Listening Port (Max)** and **Listening Port (Min)** fields.
Default Data Connection Method	Specifies the default method used to set up data connections with the FTP server. Select **Proxy Sends PASV** to send a PASV to the FTP server and let the FTP server open a listening port. Select **Proxy Sends PORT** to set up a listening port on the Traffic Edge side of the connection first.
Shared Server Connections	When enabled, server control connections can be shared between multiple anonymous FTP clients.
	Cacheability
FTP Caching	Enables or disables FTP caching. When enabled, Traffic Edge serves FTP requests from FTP clients from the cache. When disabled, Traffic Edge acts as a proxy server and forwards all FTP requests from FTP clients directly to the FTP server.

Directory Caching: Simple	When enabled, Traffic Edge caches directory listings without arguments: for example, `dir`/`ls`.
Directory Caching: Full	When enabled, Traffic Edge caches directory listings with arguments: for example, `ls -al`, `ls *.txt`.
Freshness: Login Information	Specifies how long (in seconds) the 220/230 responses (login messages) can stay fresh in the cache. The default value is 604800 seconds (7 days).
Freshness: Directory Listings	Specifies how long (in seconds) directory listings can stay fresh in the cache. The default value is 86400 seconds (one day).
Freshness: Files	Specifies how long (in seconds) FTP files can stay fresh in the cache. The default value is 259200 seconds (3 days).
Timeouts	
Keep-Alive Timeout: Server Control	Specifies the timeout value when the FTP server control connection is not used by any FTP clients. The default value is 90 seconds.
Inactivity Timeouts: Client Control	Specifies how long FTP client control connections can remain idle. The default value is 900 seconds.
Inactivity Timeouts: Server Control	Specifies how long the FTP server control connection can remain idle. The default value is 120 seconds.
Active Timeouts: Client Control	Specifies the how long FTP client control connections can remain open. The default value is 14400 seconds.
Active Timeouts: Server Control	Specifies how long the FTP server control connection can remain open. The default value is 14400 seconds.

Streaming Media

The **Streaming Media** configuration options are divided into the following categories:

- *Shared Settings*, described on *page 318*
- *QuickTime*, described on *page 318*
- *Real Networks*, described on *page 318*
- *Windows Media*, described on *page 318*

The **Streaming Media** configuration options display only if you have installed Traffic Edge Media Extension and you have enabled QuickTime, Real Networks, and/or Windows Media.

Shared Settings

The following table describes the **Shared Setting** configuration options.

Option	Description
	Common
RTSP Proxy Port	Specifies the proxy port used for RTSP requests shared by Real Networks and QuickTime. The default port is 554. If you change this option, you must restart Traffic Edge.

QuickTime

The following table describes the **QuickTime** configuration options

If you change any of the **QuickTime** options, you must restart Traffic Edge.

Option	Description
	Media Bridge
Media Bridge Name	Specifies the name of the MediaBridge node.
Media Bridge Port	Specifies the port number of the MediaBridge node. The default port is 10036.
Media Bridge Mount Point	Specifies the mount point for MDN streams.
Monitor Name	Specifies the name of the monitoring agent.
Monitor Port	Specifies the port of the monitoring agent. The default port is 10088.

Real Networks

The following table describes the **Real Networks** configuration options.

Option	Description
	General
RealProxy Restart Limit	Specifies the number of seconds that the `traffic_cop` process waits before restarting RealProxy. The default value is 20 seconds. If you change this option, you must restart Traffic Edge.

Windows Media

The following table describes the **Windows Media** configuration options.

If you change any of the **Windows Media** options, you must restart Traffic Edge.

Option	Description
	General
Windows Media Proxy Port	Specifies the proxy port used for MMS connections.
ASX Rewrite	Enables or disables the ASX rewrite option, which configures Traffic Edge to rewrite `.asx` files to point to Traffic Edge instead of the origin server.

Retransmit Window: Maximum Memory Size	Specifies how much memory Traffic Edge allocates to store data to reply to retransmit requests. The default value is 20971520 bytes (20 MB).
Multicast	
Multicast	Enables or disables Traffic Edge multicasting for Windows Media. Refer to *Using WMT Multicast, on page 80*.
Media Push	
Media Push	Enables or disables the media push option that lets you preload WMT media files into the Traffic Edge cache. Refer to *Using WMT Media Push, on page 76*.
Port	Specifies the port used for media push. The default port is 1900.
Password	Specifies the password used for media push.
Media Bridge	
Media Bridge Name	Specifies the name of the MediaBridge node.
Media Bridge Port	Specifies the port number of the MediaBridge node. The default port is 10022.
Media Bridge Mount Point	Specifies the mount point for MDN streams.
Monitor Name	Specifies the name of the monitoring agent.
Monitor Port	Specifies the port of the monitoring agent. The default port is 10088.

Content Routing

The **Content Routing** configuration options are divided into the following categories:

- *Hierarchies*, described below
- *Reverse Proxy* described on *page 322*
- *Mapping and Redirection*, described on *page 323*
- *Browser Auto-Config*, described on *page 325*

Hierarchies

The following table describes the **Hierarchy** configuration options.

Option	Description
Parenting	
Parent Proxy	Enables or disables the parent caching option. When enabled, Traffic Edge can participate in a cache hierarchy. You can point your Traffic Edge at a parent cache (either another Traffic Edge or a different caching product) to form a hierarchy in which a child cache relies upon a parent in fulfilling client requests. Refer to *Parent Caching, on page 159*).
No DNS and Just Forward to Parent	When enabled (and if parent caching is enabled), Traffic Edge does not perform DNS lookups on request hostnames.

Parent Proxy Cache Rules	Displays a table listing the rules in the `parent.config` file that identify the parent proxies used in a hierarchy and configure selected URL requests to bypass parent proxies.
Refresh	Updates the table to display the most up-to-date rules in the `parent.config` file. Click this button after you have added or modified rules with the configuration file editor.
Edit File	Opens the configuration file editor so that you can edit and add rules to the `parent.config` file. The configuration file editor page is described below.
parent.config Configuration File Editor	
rule display box	Lists the `parent.config` file rules. You must select a rule to edit it. The buttons on the left of the box allow you to delete or move the selected rule up or down in the list.
Add	Adds a new rule to the rule display box at the top of the configuration file editor page. You must enter information in the fields provided before you click this button.
Set	Updates the rule display box at the top of the configuration file editor page with any configuration changes you make**.** You must select a rule and change its properties before you click this button.
Primary Destination Type	Lists the primary destination types: **dest_domain** is a requested domain name. **dest_host** is a requested hostname. **dest_ip** is a requested IP address. **url_regex** is a regular expression to be found in a URL.
Primary Destination Value	Specifies the value of the primary destination type; for example, if the primary destination type is **dest_ip**, the value for this field can be 123.456.78.9.
Parent Proxies	Specifies the IP addresses or hostnames of the parent proxies and the port numbers used for communication. If the request cannot be handled by the last parent server in the list, it will be routed to the origin server. Separate each entry with a semicolon: for example, parent1:8080; parent2:8080.
Round Robin	Select **true** if you want Traffic Edge to go through the parent cache list in a round-robin based on client IP address. Select **strict** if you want Traffic Edge to serve requests strictly in turn; for example, machine `proxy1` serves the first request, `proxy2` serves the second request, and so on. Select **false** if you do not want round-robin selection to occur.
Go direct	Select **true** if you want requests to bypass parent hierarchies and go directly to the origin server. Select **false** if you do n*ot* want requests to bypass parent hierarchies.
Secondary Specifiers: Time	Specifies a time range, such as 08:00-14:00.
Secondary Specifiers: Prefix	Specifies a prefix in the path part of a URL.

Secondary Specifiers: Suffix	Specifies a file suffix in the URL.
Secondary Specifiers: Source IP	Specifies the IP address of the client.
Secondary Specifiers: Port	Specifies the port in a requested URL. You can specify a single port or a range of ports.
Secondary Specifiers: Method	Specifies a request URL method.
Secondary Specifiers: Scheme	Specifies the protocol of a requested URL.
Secondary Specifiers: MIXT Scheme	Specifies the media protocol type of a requested URL.
Clear Fields	Clears all the fields provided.
Apply	Applies the configuration changes. If you set invalid rules, Traffic Edge displays a warning message when you click the **Apply** button.
Close	Exits the configuration file editor. Important: You must click the **Apply** button before you click **Close**; otherwise, all configuration changes will be lost.
ICP Peering	
ICP Mode	Specifies the ICP mode for ICP peering: Select **Only Receive Queries** to configure Traffic Edge to receive ICP queries only. Select **Send/Receive Queries** to configure Traffic Edge to both send and receive ICP queries. Select **Disabled** to turn off ICP peering. Refer to *ICP Peering, on page 163*, for more information.
ICP Port	Specifies the port Traffic Edge uses for ICP messages. The default port is 3130.
ICP Multicast	Enables or disables ICP multicast. Select **Enabled**, to send ICP messages through multicast if your Traffic Edge has a multicast channel connection to its peers.
ICP Query Timeout	Specifies the timeout for ICP queries. The default is 2 seconds.
ICP Peers	Displays a table listing the rules in the `icp.config` file that specify the ICP peers (parent and sibling caches).
Refresh	Updates the table to display the most up-to-date rules in the `icp.config` file. Click this button after you have added or modified rules with the configuration file editor.
Edit File	Opens the configuration file editor so that you can edit and add rules to the `icp.config` file. The configuration file editor page is described below.
icp.config Configuration File Editor	
rule display box	Lists the `icp.config` file rules. You must select a rule to edit it. The buttons on the left of the box allow you to delete or move the selected rule up or down in the list.
Add	Adds a new rule to the rule display box at the top of the configuration file editor page. You must enter information in the fields provided before you click this button.

Set	Updates the rule display box at the top of the configuration file editor page with any configuration changes you make**.** You must select a rule and change its properties before you click this button.
Peer Hostname	Specifies the hostname of the ICP Peer. This field is optional if the IP address of the ICP peer is specified in the **Peer IP** field below.
Peer IP	Specifies the IP address of the ICP Peer. This field is optional if the hostname of the ICP peer is specified in the **Peer Hostname** field above.
Peer Type	Specifies the type of ICP peer: parent or sibling.
Proxy Port	Specifies the port number of the TCP port used by the ICP peer for proxy communication.
ICP Port	Specifies the port number of the UDP port used by the ICP peer for ICP communication.
Multicast	Enables or disables multicast mode.
Multicast IP	Specifies the multicast IP address if multicast is enabled.
Multicast TTL	Specifies the multicast time to live: Select **single subnet** if IP multicast datagrams are not forwarded beyond a single subnetwork. Select **multiple subnets** to allow delivery of IP multicast datagrams to more than one subnet (if there are one or more multicast routers attached to the first hop subnet).
Clear Fields	Clears all the fields provided.
Apply	Applies the configuration changes. If you set invalid rules, Traffic Edge displays a warning message when you click the **Apply** button.
Close	Exits the configuration file editor. Important: You must click the **Apply** button before you click **Close**; otherwise, all configuration changes will be lost.

Reverse Proxy

The following table describes the **Reverse Proxy** configuration options.

Option	Description
	General
Reverse Proxy	Enables or disables reverse proxy for HTTP and streaming media requests. As a reverse proxy, Traffic Edge serves requests on behalf of origin servers. Traffic Edge is configured to be the origin server the user is trying to connect to (typically, the advertised hostname of the origin server resolves to Traffic Edge, which is acting as the real origin server). For more information about reverse proxy, Refer to *Chapter 7, Reverse Proxy and HTTP Redirects*.

FTP	
Reverse Proxy	Enables or disables FTP reverse proxy. As a reverse proxy, Traffic Edge serves requests on behalf of FTP servers. Refer to *FTP Reverse Proxy, on page 138*.

Mapping and Redirection

The following table describes the **Mapping and Redirection** configuration options.

Option	Description
Serve Mapped Hosts Only	Select **Required** if you want Traffic Edge to serve requests only to origin servers listed in the mapping rules of the `remap.config` file. If a request does not match a rule in the `remap.config` file, the browser receives an error. This option provides added security for your Traffic Edge system.
Retain Client Host Header	When enabled, Traffic Edge retains the client host header in a request (it does not include the client host header in the mapping translation).
Redirect No-Host Header to URL	Specifies the alternate URL to which to direct incoming requests from older clients that do not provide a `Host:` header. Inktomi recommends that you set this option to a page that explains the situation to the user and advises a browser upgrade or provides a link directly to the origin server, bypassing the Traffic Edge. Alternatively, you can specify a map rule that maps requests without `Host:` headers to a particular server.
URL Remapping Rules	Displays a table listing the mapping rules in the `remap.config` file so that you can perform the following actions: - Map URL requests for a specific origin server to the appropriate location on Traffic Edge when Traffic Edge acts as a reverse proxy for that particular origin server. - Reverse-map server location headers so that when origin servers respond to a request with a location header that redirects the client to another location, the clients do not bypass the Traffic Edge. - Redirect HTTP requests permanently or temporarily without Traffic Edge having to contact any origin servers.
Refresh	Updates the table to display the most up-to-date rules in the `remap.config` file. Click this button after you have added or modified rules with the configuration file editor.
Edit File	Opens the configuration file editor so that you can edit and add rules to the `remap.config` file. The configuration file editor page is described below.

remap.config Configuration File Editor	
rule display box	Lists the `remap.config` file rules. You must select a rule to edit it. The buttons on the left of the box allow you to delete or move the selected rule up or down in the list.
Add	Adds a new rule to the rule display box at the top of the configuration file editor page. You must enter information in the fields provided before you click this button.
Set	Updates the rule display box at the top of the configuration file editor page with any configuration changes you make**.** You must select a rule and change its properties before you click this button.
Rule Type	Lists the type of rules you can add to the `remap.config` file: **map** translates an incoming request URL to the appropriate origin server URL (reverse proxy). **reverse_map** translates the URL in origin server redirect responses to point to the Traffic Edge (reverse proxy). **redirect** redirects HTTP requests permanently without having to contact the origin server. Permanent redirects notify the browser of the URL change (by returning an HTTP status code 301) so that the browser can update bookmarks. **redirect_temporary** redirects HTTP requests temporarily without having to contact the origin server. Temporary redirects notify the browser of the URL change for the current request only (by returning an HTTP status code 307).
From Scheme	Specifies the protocol of the URL to map from.
From Host	Specifies the hostname of the URL to map from.
From Port (Optional)	Specifies the port number in the URL to map from.
From Path Prefix (Optional)	Specifies the path prefix of the URL to map from.
To Scheme	Specifies the protocol of the URL to map to.
To Host	Specifies the hostname of the URL to map to.
To Port (Optional)	Specifies the port number of the URL to map to.
To Path Prefix (Optional)	Specifies the path prefix of the URL to map to.
MIXT Scheme	Specifies the media protocol type of the mapping rule: **QT** for QuickTime or **RNI** for Real Networks.
Clear Fields	Clears all the fields provided.
Apply	Applies the configuration changes. If you set invalid rules, Traffic Edge displays a warning message when you click the **Apply** button.
Close	Exits the configuration file editor. Important: You must click the **Apply** button before you click **Close**; otherwise, all configuration changes will be lost.

FTP	
FTP Remapping Rules	Displays a table listing the FTP mapping rules in the `ftp_remap.config` file. Traffic Edge uses the rules to direct any incoming FTP requests to the FTP server if the requested document is a cache miss or is stale.
Refresh	Updates the table to display the most up-to-date rules in the `ftp_remap.config` file. Click this button after you have added or modified rules with the configuration file editor.
Edit File	Opens the configuration file editor so that you can edit and add rules to the `ftp_remap.config` file. The configuration file editor page is described below.
ftp_remap.config Configuration File Editor	
rule display box	Lists the `ftp_remap.config` file rules. You must select a rule to edit it. The buttons on the left of the box allow you to delete or move the selected rule up or down in the list.
Add	Adds a new rule to the rule display box at the top of the configuration file editor page. You must enter information in the fields provided before you click this button.
Set	Updates the rule display box at the top of the configuration file editor page with any configuration changes you make**.** You must select a rule and change its properties before you click this button.
Proxy Hostname	Specifies the IP address or hostname assigned to Traffic Edge.
Proxy Port	Specifies the proxy port used by Traffic Edge to listen to FTP traffic.
FTP Server	Specifies the IP address or hostname of the FTP server to which you want to redirect requests.
FTP Server Port	Specifies the port number used by the FTP server to handle requests.
Clear Fields	Clears all the fields provided.
Apply	Applies the configuration changes. If you set invalid rules, Traffic Edge displays a warning message when you click the **Apply** button.
Close	Exits the configuration file editor. Important: You must click the **Apply** button before you click **Close**; otherwise, all configuration changes will be lost.

Browser Auto-Config

The following table describes the **Browser Auto-Config** options.

Option	Description
PAC	
Auto-Configuration Port	Specifies the port Traffic Edge uses to download the autoconfiguration file to browsers. The port cannot be assigned to any other process. The default port is 8083. If you change this option, you must restart Traffic Edge.
PAC Settings	Lets you edit the PAC file (`proxy.pac`). Refer to *Using a PAC File, on page 86*.

WPAD	
WPAD Settings	Lets you edit the `wpad.dat` file. Refer to *Using WPAD, on page 88*.

Security

The **Security** configuration options are divided into the following categories:

- *Connection Control*, described below
- *Access Control*, described on *page 328*
- *SSL Termination*, described on *page 331*
- *SOCKS*, described on *page 333*

Connection Control

The following table describes the **Connection Control** configuration options.

Option	Description
Proxy Access	
Access Control	Displays the rules in the `ip_allow.config` file that control which clients can access to the Traffic Edge proxy cache. By default, all remote hosts are allowed to access the Traffic Edge proxy cache.
Refresh	Updates the table to display the most up-to-date rules in the `ip_allow.config` file. Click this button after you have added or modified rules with the configuration file editor.
Edit File	Opens the configuration file editor so that you can edit and add rules to the `ip_allow.config` file. The configuration file editor page is described below.
ip_allow.config Configuration File Editor	
rule display box	Lists the `ip_allow.config` file rules. You must select a rule to edit it. The buttons on the left of the box allow you to delete or move the selected rule up or down in the list.
Add	Adds a new rule to the rule display box at the top of the configuration file editor page. You must enter information in the fields provided before you click this button.
Set	Updates the rule display box at the top of the configuration file editor page with any configuration changes you make**.** You must select a rule and change its properties before you click this button.
IP Action	Lists the type of rules you can add. An **ip_allow** rule allows the clients listed in the **Source IP** field to access the Traffic Edge proxy cache. An **ip_deny** rule denies the clients listed in the **Source IP** field access to the Traffic Edge proxy cache.

Source IP	Specifies the IP address or range of IP addresses of the clients that are allowed access or are denied access to the Traffic Edge proxy cache.
Clear Fields	Clears all the fields provided.
Apply	Applies the configuration changes. If you set invalid rules, Traffic Edge displays a warning message when you click the **Apply** button.
Close	Exits the configuration file editor. Important: You must click the **Apply** button before you click **Close**; otherwise, all configuration changes will be lost.
ARM Security	
ARM Security	Enables or disables the ARM security option that restricts the type of communication possible with machine running Traffic Edge. For information about the ARM Security option, refer to *Controlling Host Access to the Traffic Edge Machine, on page 204*. If you change this option, you must restart Traffic Edge.
Access Control List	Displays a table listing the rules in the `arm_security.config` file that restrict the type of communication possible with machines running Traffic Edge. For more details about the ARM security feature, refer to*Controlling Host Access to the Traffic Edge Machine, on page 204*.
Refresh	Updates the table to display the most up-to-date rules in the `arm_security.config` file. Click this button after you have added or modified rules with the configuration file editor.
Edit File	Opens the configuration file editor so that you can edit and add rules to the `arm_security.config` file. The configuration file editor page is described below.
arm_security.config Configuration File Editor	
rule display box	Lists the `arm_security.config` file rules. You must select a rule to edit it. The buttons on the left of the box allow you to delete or move the selected rule up or down in the list.
Add	Adds a new rule to the rule display box at the top of the configuration file editor page. You must enter information in the fields provided before you click this button.
Set	Updates the rule display box at the top of the configuration file editor page with any configuration changes you make**.** You must select a rule and change its properties before you click this button.
Rule Type	Specifies the rule type: An **open** rule specifies the ports that are open by default, for either TCP or UDP. You must specify the ports you want to open in the **Open Port** field. A **deny** rule specifies the hosts that are denied access to specific destination ports, for either TCP or UDP. An **allow** rule specifies the hosts that are allowed access to specific destination ports, for either TCP or UDP.
Connection Type	Specifies the type of connection used: TCP or UDP.
Source IP	Specifies the IP address or range of IP addresses of the source of the communication.

Source Port	Specifies the source port or range of source ports from which TCP traffic is allowed.
Destination IP	Specifies the IP address or range of IP addresses of the destination of the communication.
Destination Port	Specifies the destination port or range of destination ports from which TCP traffic is allowed or denied.
Open Port	Specifies the port or series of ports that are open by default.
Clear Fields	Clears all the fields provided.
Apply	Applies the configuration changes. If you set invalid rules, Traffic Edge displays a warning message when you click the **Apply** button.
Close	Exits the configuration file editor. Important: You must click the **Apply** button before you click **Close**; otherwise, all configuration changes will be lost.

Access Control

The following table describes the **Access Control** configuration options.

Option	Description
	Filtering
Filtering	Displays a table listing the rules in the `filter.config` file that deny or allow particular URL requests, strip header information from client requests, and specify LDAP, NTLM, and RADIUS authentication rules and NTLM authorization rules.
Refresh	Updates the table to display the most up-to-date rules in the `filter.config` file. Click this button after you have added or modified rules with the configuration file editor.
Edit File	Opens the configuration file editor so that you can edit and add rules to the `filter.config` file. The configuration file editor page is described below.
filter.config Configuration File Editor	
rule display box	Lists the `filter.config` file rules. You must select a rule to edit it. The buttons on the left of the box allow you to delete or move the selected rule up or down in the list.
Add	Adds a new rule to the rule display box at the top of the configuration file editor page. You must enter information in the fields provided before you click this button.
Set	Updates the rule display box at the top of the configuration file editor page with any configuration changes you make**.** You must select a rule and change its properties before you click this button.

Rule Type	Specifies the rule type: Select **allow** to allow particular URL requests; Traffic Edge caches and serves the requested content. Select **deny** to deny requests for objects from specific destinations. When a request is denied, the client receives an access denied message. Select **ldap** to specify authentication rules that determine which users must be authenticated to access particular sites on the Internet and which LDAP servers are used. Select **ntlm** to specify authentication rules that determine which users must be authenticated to access particular sites on the Internet and which domain controllers are used. Select **ntlm** also to specify NTLM authorization rules. Select **radius** to specify authentication rules that determine which users must be authenticated to access particular sites on the Internet and which RADIUS servers are used. Select **strip_hdr** to specify which client request header information you want to strip.
Primary Destination Type	Lists the primary destination types: **dest_domain** is a requested domain name. **dest_host** is a requested hostname. **dest_ip** is a requested IP address. **url_regex** is a regular expression to be found in a URL.
Primary Destination Value	Specifies the value of the primary destination type; for example, if the primary destination type is **dest_ip**, the value for this field can be 123.456.78.9.
Header Type	Specifies the client request header information that you want to keep or strip. This option applies to a **strip_hdr** rule only.
Secondary Specifiers (optional): Time	Specifies a time range, such as 08:00-14:00.
Secondary Specifiers (optional): Prefix	Specifies a prefix in the path part of a URL.
Secondary Specifiers (optional): Suffix	Specifies a file suffix in the URL.
Secondary Specifiers (optional): Source IP	Specifies the IP address of the client sending the request. This secondary specifier is not supported for PNA content.
Secondary Specifiers (optional): Port	Specifies the port in a requested URL. You can specify a single port or a range of ports.
Secondary Specifiers (optional): Method	Specifies a request URL method.
Secondary Specifiers (optional): Scheme	Specifies the protocol of a requested URL.
Secondary Specifiers (optional): MIXT Scheme	Specifies the media protocol type of a requested URL: **QT** for QuickTime or **RNI** for Real Networks.
Authentication and Authorization Specifiers: Server Name	Specifies the LDAP server name for LDAP authentication or the domain controller for NTLM authorization.

Authentication and Authorization Specifiers: Base Distinguished Name	Specifies the LDAP or NTLM Base Distinguished Name.
Authentication and Authorization Specifiers: UID filter	Specifies the LDAP or NTLM UID filter.
Authentication and Authorization Specifiers: Attribute Name	Specifies the LDAP or NTLM attribute name.
Authentication and Authorization Specifiers: Attribute Value	Specifies the LDAP or NTLM attribute value.
Authentication and Authorization Specifiers: Bind DN	Specifies the DN of an authorized user. This option applies to only LDAP authentication rules and NTLM authorization rules.
Authentication and Authorization Specifiers: Bind Password	Specifies the password used when binding to the LDAP server or Domain Controller. You must click the **Apply Password** button to apply a new password rule based on the values currently specified.
Authentication and Authorization Specifiers: Realm (optional)	Specifies the realm for LDAP, NTLM, and RADIUS rules. The default value is Traffic Edge. Important: For WMT (HTTP streaming), do not enter a value that contains the string `server`: otherwise, proxy authentication with the Windows Media Server does not work correctly.
Authentication and Authorization Specifiers: Redirect URL (optional)	Specifies the URL that Traffic Edge redirects to when an error occurs for LDAP, NTLM, or RADIUS rules.
Clear Fields	Clears all the fields provided.
Apply	Applies the configuration changes. If you set invalid rules, Traffic Edge displays a warning message when you click the **Apply** button.
Close	Exits the configuration file editor. Important: You must click the **Apply** button before you click **Close**; otherwise, all configuration changes will be lost.

LDAP

The LDAP configuration options display on the **Configure** tab only if you have enabled LDAP.

If you change any of the **LDAP** options, you must restart Traffic Edge.

Purge Cache on Authentication Failure	When enabled, Traffic Edge deletes the authorization entry for the client in the Traffic Edge authentication cache if authorization fails. Note: If you change this option, you must restart Traffic Edge.
LDAP Server: Hostname	Specifies the hostname of the LDAP server.
LDAP Server: Port	Specifies the port used for LDAP communication. The default port number is 389.
LDAP Base Distinguished Name	Specifies the base Distinguished Name (DN). You can obtain this value from your LDAP administrator. Important: You must specify a correct base DN; otherwise, LDAP authentication fails to operate.

Radius	
The **Radius** configuration options display on the **Configure** tab only if you have enabled **Radius**. If you change any of the **Radius** options, you must restart Traffic Edge.	
Primary Radius Server: Hostname	Specifies the hostname or IP address of the primary RADIUS authentication server.
Primary Radius Server: Port	Specifies the port that Traffic Edge uses to communicate with the primary RADIUS authentication server. The default port is 1812.
Primary Radius Server: Shared Key	Specifies the key to use for encoding.
Secondary Radius Server (Optional): Hostname	Specifies the hostname or UP address of the secondary RADIUS authentication server.
Secondary Radius Server (Optional): Port	Specifies the port that Traffic Edge uses to communicate with the secondary RADIUS authentication server. The default port is 1812.
Secondary Radius Server (Optional): Shared Key	Specifies the key to use for encoding.
NTLM	
The **NTLM** configuration options display on the **Configure** tab only if you have enabled NTLM. If you change any of the **NTLM** options, you must restart Traffic Edge.	
Domain Controller Hostnames	Specifies the hostnames of the domain controllers. You must separate each entry with a comma: for example, `host1`, `host2`, `host3`.
NT Domain Name	Specifies the NT domain name against which Traffic Edge should authenticate.
Load Balancing	Enables or disables load balancing. When enabled, Traffic Edge balances the load when sending authentication requests to the domain controllers.

SSL Termination

The following table describes the **SSL Termination** configuration options.

The **SSL Termination** configuration options display on the **Configure** tab only if you have enabled SSL Termination.

Option	Description
General	
SSL Termination Port	Specifies the port used for SSL communication. The default port is 443. If you change this option, you must restart Traffic Edge.
Accelerator	
SSL Accelerator Type	
None - Software Algorithms	When selected, Traffic Edge does not use an accelerator card. The CPU of the Traffic Edge machine determines the number of requests served per second. If you change this option, you must restart Traffic Edge.
NCipher Nfast Accelerator Card: Selected	Select this option if the nCipher nFast accelerator card is installed on your Traffic Edge machine. If you change this option, you must restart Traffic Edge.

NCipher Nfast Accelerator Card: Library Path	Specifies the library path for the nCipher nFast accelerator card. You need only change this option if you did not use the default path when you installed the card. If you change this option, you must restart Traffic Edge.
Rainbow Crypto Swift Accelerator Card: Selected	Select this option if the Rainbow CryptoSwift accelerator card is installed on your Traffic Edge machine. If you change this option, you must restart Traffic Edge.
Rainbow Crypto Swift Accelerator Card: Library Path	Specifies the library path for the Rainbow CryptoSwift accelerator card. Note: You need only change this option if you did not use the default path when you installed the card. If you change this option, you must restart Traffic Edge.
Compaq Atalla Accelerator Card: Selected	Select this option if the Compaq Atalla accelerator card is installed on your Traffic Edge machine. If you change this option, you must restart Traffic Edge.
Compaq Atalla Accelerator Card: Library Path	Specifies the library path for the Compaq Atalla accelerator card. You need only change this option if you did not use the default path when you installed the card. If you change this option, you must restart Traffic Edge.
	Client-Proxy
Client Certificate	Specifies if Traffic Edge requires client certificates for authentication: Select **Not Required** if no client certificates are required. Traffic Edge does not verify client certificates during the SSL handshake. Access to Traffic Edge depends on Traffic Edge configuration options such as access control lists. Select **Optional** if client certificates are optional. If a client has a certificate, the certificate is validated. If the client does not have a certificate, the client is still allowed access to Traffic Edge unless access is denied through other Traffic Edge configuration options. Select **Required** if client certificates are required. The client must be authenticated during the SSL handshake. Clients without a certificate are not allowed to access Traffic Edge. If you change this option, you must restart Traffic Edge.
Server Certificate File	Specifies the filename of the Traffic Edge SSL server certificate. Traffic Edge provides a demo server certificate called `server.pem` in the `config` directory. You can use this certificate to verify that the SSL feature is working. If you are using multiple server certificates, set this option to specify the default filename. If you change this option, you must restart Traffic Edge.
Server Private Key	Specifies the filename of the Traffic Edge private key. Change this option only if the private key is not located in the Traffic Edge SSL server certificate file. If you change this option, you must restart Traffic Edge.

Certificate Authority	Specifies the filename of the Certificate Authority (CA) that client certificates will be verified against. If you change this option, you must restart Traffic Edge.
SSL Multi-Certificate	Displays the `ssl_multicert.config` file that allows an SSL certificate and a private key to be tied to a specific IP address on a multihomed machine. Refer to *ssl_multicert.config, on page 454*.
Proxy-Server	
Certificate Verification	Specifies if Traffic Edge needs to verify the origin server certificate with the CA. If you change this option, you must restart Traffic Edge.
Client Certificate File	Specifies the filename of the client certificate if you have installed an SSL client certificate on Traffic Edge. If you change this option, you must restart Traffic Edge
Client Private Key	Specifies the filename of the Traffic Edge private key. Change this option only if the private key is not located in the Traffic Edge SSL client certificate file. If you change this option, you must restart Traffic Edge.
Certificate Authority	Specifies the filename of the certificate authority against which the origin server will be verified. If you change this option, you must restart Traffic Edge.

SOCKS

The following table describes the **SOCKS** configuration options.

The **SOCKS** configuration options display on the **Configure** tab only if you have enabled SOCKS.

Option	Description
General	
SOCKS Version	Specifies the version of SOCKS used on your SOCKS server. Traffic Edge supports SOCKS Version 4 and Version 5. If you change this option, you must restart Traffic Edge.
Proxy	
SOCKS Proxy	Enables or disables the SOCKS Proxy option. As a SOCKS proxy, Traffic Edge accepts SOCKS traffic (usually on port 1080). Traffic Edge detects and serves HTTP requests but forwards all other requests directly to the SOCKS server. For more information about the SOCKS Proxy option, refer to *Configuring SOCKS Firewall Integration, on page 210*. If you change this option, you must restart Traffic Edge.
SOCKS Proxy Port	Specifies the port on which Traffic Edge accepts SOCKS traffic. This is usually port 1080. If you change this option, you must restart Traffic Edge.

Server	
SOCKS Server: Default Servers	Specifies the names and the ports of the default SOCKS servers with which Traffic Edge communicates. Each entry must be separated by a semicolon (;): for example, `socks1:1080;socks2:4080` If you change this option, you must restart Traffic Edge. You can perform additional SOCKS server configuration in the `socks.config` described below. You can specify that requests to specific origin servers go through specific SOCKS servers.
Socks Server Rules	Displays a table listing the rules in the `socks.config` file that control the SOCKS servers through which Traffic Edge must go to access specific origin servers and the order in which Traffic Edge goes through the SOCKS server list. You can also specify the origin servers that you want Traffic Edge to access directly without going through the SOCKS server, and the username and password used by Traffic Edge to connect to a SOCKS Version 5 server.
Refresh	Updates the table to display the most up-to-date rules in the `socks.config` file. Click this button after you have added or modified rules with the configuration file editor.
Edit File	Opens the configuration file editor so that you can edit and add rules to the `socks.config` file. The configuration file editor page is described below.
socks.config Configuration File Editor	
rule display box	Lists the `socks.config` file rules. You must select a rule to edit it. The buttons on the left of the box allow you to delete or move the selected rule up or down in the list.
Add	Adds a new rule to the rule display box at the top of the configuration file editor page. You must enter information in the fields provided before you click this button.
Set	Updates the rule display box at the top of the configuration file editor page with any configuration changes you make**.** You must select a rule and change its properties before you click this button.
Rule Type	Specifies the rule type: Select **no_socks** to specify the origin servers you want Traffic Edge to access directly without going through the SOCKS server. Enter the IP addresses of the origin servers in the **Destination IP** field. Select **auth** to specify the username and password Traffic Edge uses for authentication with a SOCKS Version 5 server. Enter the username in the **Username** field and the password in the **Password** field. Select **multiple_socks** to specify the SOCKS servers through which Traffic Edge must go to reach specific origin servers. Enter the hostnames or the IP addresses of the SOCKS servers in the **SOCKS Servers** field and the IP addresses of the origin servers in the **Destination IP** field. Select how strict Traffic Edge should follow round robin from the **Round Robin** drop-down list.

Username	Specifies the username Traffic Edge must use to authenticate with a SOCKS Version 5 server. This field applies to an **auth** rule type only.
Password	Specifies the password Traffic Edge must use to authenticate with a SOCKS Version 5 server. This field applies to an **auth** rule type only.
Destination IP	For a **multiple_socks** rule, specify either a single IP address *or* a range of IP addresses of the origin servers with which Traffic Edge must use the SOCKS servers specified in the **SOCKS Servers** field below. For a **no_socks** rule, specify the IP addresses of the origin servers that you want Traffic Edge to access directly (without going through the SOCKS server). You can enter a single IP address, a range of IP addresses, or a list of IP addresses. Separate each entry in the list with a comma.
SOCKS Servers	Specify the hostnames or the IP addresses of the SOCKS servers you want to go through to service requests for the origin servers listed in the **Destination IP** field above. Separate each entry with a semicolon. This field applies to a **multiple_socks** rule type only.
Round Robin	Specifies how strict Traffic Edge should follow round robin. You can select strict, or false. This field applies to a **multiple_socks** rule type only.
Clear Fields	Clears all the fields provided.
Apply	Applies the configuration changes. If you set invalid rules, Traffic Edge displays a warning message when you click the **Apply** button.
Close	Exits the configuration file editor. Important: You must click the **Apply** button before you click **Close**; otherwise, all configuration changes will be lost.

Subsystems

The **Subsystem** configuration options are divided into the following categories:

- *Cache*, described below
- *Logging*, described on *page 338*

Cache

The following table describes the **Cache** configuration options.

Option	Description
	General
Allow Pinning	Enables or disables the cache pinning option, which lets you keep objects in the cache for a specified time. You set cache pinning rules in the `cache.config` file (refer to *cache.config, on page 369*).

Ram Cache Size	Specifies the size of the RAM cache, in bytes. Traffic Edge maintains a small RAM cache of extremely popular objects. This RAM cache serves the most popular objects as fast as possible and reduces load on disks, especially during temporary traffic peaks. A value of -1 means that the RAM cache is automatically sized at approximately one megabyte per gigabyte of disk. If you change this option, you must restart Traffic Edge.
Maximum Object Size	Specifies the maximum size allowed for objects in the cache. A value of 0 (zero) means that there is no size restriction.
Partition	
Cache Partition	Displays a table showing the rules in `partition.config` file that controls how the cache is partitioned.
Refresh	Updates the table to display the most up-to-date rules in the `partition.config` file. Click this button after you have added or modified rules with the configuration file editor.
Edit File	Opens the configuration file editor so that you can edit and add rules to the `partition.config` file. The configuration file editor page is described below.
partition.config Configuration File Editor	
rule display box	Lists the `partition.config` file rules. You must select a rule to edit it. The buttons on the left of the box allow you to delete or move the selected rule up or down in the list.
Add	Adds a new rule to the rule display box at the top of the configuration file editor page. You must enter information in the fields provided before you click this button.
Set	Updates the rule display box at the top of the configuration file editor page with any configuration changes you make**.** You must select a rule and change its properties before you click this button.
Partition Number	Specifies a partition number between 1 and 255.
Scheme	Specifies the content type stored in the partition. All streaming QuickTime and WMT media content is stored in the mixt partition and all other content is stored in the http partition. Important: Traffic Edge does not use the cache to store streams for Real Networks requests but uses the filesystem instead. Refer to the Traffic Edge Installation Guide for more information.
Partition Size	Specifies the amount of cache space allocated to the partition which can be either a percentage of the total cache space or an absolute value in megabytes.
Partition Size Format	Specifies the format of the partition size: percentage or absolute.
Clear Fields	Clears all the fields provided.
Apply	Applies the configuration changes. If you set invalid rules, Traffic Edge displays a warning message when you click the **Apply** button.

Close	Exits the configuration file editor. Important: You must click the **Apply** button before you click **Close**; otherwise, all configuration changes will be lost.
Hosting	
Cache Hosting	Displays a table listing the rules in the `hosting.config` file that controls which cache partitions are assigned to specific origin servers and/or domains.
Refresh	Updates the table to display the most up-to-date rules in the `hosting.config` file. Click this button after you have added or modified rules with the configuration file editor.
Edit File	Opens the configuration file editor so that you can edit and add rules to the `hosting.config` file. The configuration file editor page is described below.
hosting.config Configuration File Editor	
rule display box	Lists the `hosting.config` file rules. You must select a rule to edit it. The buttons on the left of the box allow you to delete or move the selected rule up or down in the list.
Add	Adds a new rule to the rule display box at the top of the configuration file editor page. You must enter information in the fields provided before you click this button.
Set	Updates the rule display box at the top of the configuration file editor page with any configuration changes you make**.** You must select a rule and change its properties before you click this button.
Primary Destination Type	Specifies the primary destination rule type: Select **domain** if you want to partition the cache according to domain. Select **hostname** if you want to partition the cache according to hostname
Primary Destination Value	Specifies the domain or origin server hostname whose content you want to store on a particular partition.
Partitions	Specifies the partitions on which you want to store the content that belongs to the origin server or domain specified. Separate each partition with a comma. Note: The partitions must be created already in the `partition.config` file. For more information about creating partitions, refer to *Partitioning the Cache, on page 167*.
Clear Fields	Clears all the fields provided.
Apply	Applies the configuration changes. If you set invalid rules, Traffic Edge displays a warning message when you click the **Apply** button.
Close	Exits the configuration file editor. Important: You must click the **Apply** button before you click **Close**; otherwise, all configuration changes will be lost.

Logging

The following table describes the **Logging** configuration options.

Option	Description
	General
Logging	Enables or disables event logging so that transactions are recorded into event log files and/or error log files. Select **Log Transactions and Errors** to log transactions into your selected event log files and errors in the error log files. Select **Log Transactions Only** to log transactions into your selected event log files only. Traffic Edge does not log errors in the error log files. Select **Log Errors Only** to log errors in the error log files only. Traffic Edge does not log transactions into your selected event log files. Select **Disabled** to turn off logging.
Log Directory	Specifies the path of the directory in which Traffic Edge stores event logs. The path of this directory must be the same on every node in the Traffic Edge cluster.
Log Space: Limit	Specifies the maximum amount of space (in megabytes) allocated to the logging directory for the log files. Transaction logs can consume a lot of space. You should set this limit high enough to accommodate at least a single day's worth of uncompressed transaction logs. Also, make sure that this limit is smaller than the actual space available on the partition that contains the logging directory.
Log Space: Headroom	Specifies the tolerance for the log space limit. If the **Auto-Delete Rolled Files** option is enabled, autodeletion is triggered when the amount of free space available in the logging directory is less than the headroom.
Log Rolling: Enable/Disable	Enables or disables log file rolling. To keep log files down to manageable sizes, you can roll them at regular intervals. When the Traffic Edge rolls a log file, it stops making entries in the file and adds the filename extension `.old`. The log file autodeletion feature deletes rolled log files when space allocated to logging is nearly full (when free space in the logging directory is less than the headroom).
Log Rolling: Offset Hour	Specifies the hour when log rolling takes place. You can set a time of the day in the range 0 to 23; for example, if the offset hour is 0 (midnight) and the roll interval is 6, the log files are rolled at 00:00, 06:00, noon, and 18:00.
Log Rolling: Interval	Specifies the amount of time the Traffic Edge enters data in log files before rolling them to `.old` files. The minimum value is 300 seconds (five minutes). The default value is 86400 seconds (1 day).
Log Rolling: Auto-Delete Rolled Files	Enables autodeletion of rolled log files when available space in the log directory is low. Autodeletion is triggered when the amount of free space available in the log directory is less than the **Log Space Headroom**.

Option	Description
Formats	
Squid Format: Enable/Disable	Enables or disables the Squid log format.
Squid Format: ASCII/Binary	Select **ASCII** or **Binary** as the type of log files to be created.
Squid Format: Filename	Specifies the name used for Squid log files. The default filename is `squid.log`.
Squid Format: Header	Specifies the text header you want Squid log files to contain.
Netscape Common Format: Enable/Disable	Enables or disables the Netscape Common log format.
Netscape Common Format: ASCII/Binary	Select **ASCII** or **Binary** as the type of log file to be created.
Netscape Common Format: Filename	Specifies the name used for Netscape Common log files. The default filename is `common.log`.
Netscape Common Format: Header	Specifies the text header you want Netscape Common log files to contain.
Netscape Extended Format: Enable/Disable	Enables or disables the Netscape Extended log format.
Netscape Extended Format: ASCII/Binary	Select **ASCII** or **Binary** as the type of log file to be created.
Netscape Extended Format: Filename	Specifies the name used for Netscape Extended log files. The default filename is `extended.log`.
Netscape Extended Format: Header	Specifies the text header you want Netscape Extended log files to contain.
Netscape Extended 2 Format: Enable/Disable	Enables or disables the Netscape Extended-2 log format.
Netscape Extended 2 Format: ASCII/Binary	Select **ASCII** or **Binary** as the type of log file to be created.
Netscape Extended 2 Format: Filename	Specifies the name used for Netscape Extended-2 log files. The default filename is `extended2.log`.
Netscape Extended 2 Format: Header	Specifies the text header you want Netscape Extended-2 log files to contain.
Splitting	
Split ICP Logs	When enabled, Traffic Edge records ICP transactions in a separate log file. When disabled, Traffic Edge records ICP transactions in the same log file with HTTP and FTP entries.
Split Host Logs	If Traffic Edge is running in reverse proxy mode, it can log the transactions for each mapped origin server in a separate log file. When Split Host Logs is enabled, Traffic Edge creates a separate log file for each of the hosts listed in the `log_hosts.config` file. When Split Host Logs is disabled, Traffic Edge records transactions for all hosts in the same log file.

Option	Description
	Collation
Collation Mode	Specifies the log collation mode for this Traffic Edge node. You can use the Traffic Edge log file collation feature to keep all logged information in one place. This allows you to analyze Traffic Edge as a whole rather than as individual nodes and to use a large disk that might only be located on one of the nodes in a cluster. For more information about log file collation, refer to *Collating Event Log Files, on page 257*. Select **Collation Disabled** to disable log collation on this Traffic Edge node. Select **Be a Collation Server** to configure this Traffic Edge node to be the collation server. Select **Be a Collation Client** to configure this Traffic Edge to be a collation client. A Traffic Edge configured as a collation client sends only the active standard log files, such as Squid, Netscape Common, and so on, to the collation server. If you select this option, you must enter the hostname of the collation server for your cluster in the **Log Collation Server** field. Note: When logs are collated, the source of the log entry—its node of origin—is lost unless you turn on the *Log collation host tagged* option (described below). Important: Log collation consumes cluster bandwidth in sending all log entries to a single node. It can therefore impact the performance of the cluster. If you want Traffic Edge as a collation client to send custom (XML-based) log files, you must specify a `LogObject` in the `logs_xml.config` file.
Log Collation Server	Specifies the hostname of the log collation server to which you want to send log files.
Log Collation Port	Specifies the port used for communication between the collation server and client. You must specify a port number in all cases, except when log collation is inactive. The default port number is 8085. Do not change the port number unless there is a conflict with another service already using the port.
Log Collation Secret	Specifies the password for the log collation server and the other nodes in the cluster. This password is used to validate logging data and prevent the exchange of arbitrary information.
Log Collation Host Tagged	When enabled, Traffic Edge adds the hostname of the Traffic Edge node that generated the log entry to end of the entry in the collated log file.
Log Collation Orphan Space	Specifies the maximum amount of space (in megabytes) allocated to the logging directory for storing orphan log files on the Traffic Edge node. Traffic Edge creates orphan log entries when it cannot contact the log collation server.
	Custom
Custom Logging	Enables or disables custom logging.
Custom Log File Definitions	Displays the `logs_xml.config` file so that you can configure Traffic Edge custom (XML-based) logging options. Refer to *logs_xml.config, on page 387*.

Networking

The **Networking** configuration options are divided into the following categories:

- *System*, described below
- *Connection Management*, described on *page 342*
- *ARM*, described on *page 342*
- *WCCP*, described on *page 346*
- *DNS Proxy*, described on *page 347*
- *DNS Resolver*, described on *page 347*
- *Virtual IP*, described on *page 349*

System

The following table describes the **System** configuration options.

Note The **System** configuration options are supported for UNIX only.

Option	Description
	General
Hostname	Specifies the hostname of the machine running Traffic Edge.
Gateway	Specifies the gateway for the machine running Traffic Edge.
DNS: Domain Name	Specifies the domain in which the machine running Traffic Edge is grouped.
DNS: IP Address of Primary DNS	Specifies the address of the primary DNS server used by the machine running Traffic Edge.
DNS: IP Address of Secondary DNS	Specifies the address of the secondary DNS server used by the machine running Traffic Edge.
DNS: IP Address of Third DNS	Specifies the address of the third DNS server used by the machine running Traffic Edge.
	NIC
Interface *name* Parameters: Enable/Disable	Enables or disables the network interface. You cannot disable your primary network interface.
Interface *name* Parameters: Onboot	Specifies if the network interface is enabled automatically when the Traffic Edge node boots. You cannot change this option for your primary network interface.
Interface *name* Parameters: IP Address	Specifies the IP address of the network interface.
Interface *name* Parameters: Netmask	Specifies the netmask for the network interface.
Interface *name* Parameters: Gateway	Specifies the gateway for the network interface.

CAUTION Use caution when changing the network interface parameters on the **NIC** tab. Traffic Edge does not warn you of any network activity problems caused by your modifications.

Connection Management

The following table describes the **Connection Management** configuration options.

Option	Description
	Throttling
Throttling Net Connections	Specifies the maximum number of network connections that Traffic Edge accepts. Setting a Traffic Edge throttle limit helps to prevent system overload when traffic bottlenecks develop. When network connections reach this limit, Traffic Edge queues new connections until existing connections close. Important: Do not set this variable below the minimum value of 100.
	Load Shedding
Maximum Connections	Specifies the maximum number of client connections allowed before the Traffic Edge ARM (transparent mode) starts forwarding incoming requests directly to the origin server. The default value is 1 million connections. If you change this option, you must restart Traffic Edge.
	Congestion Control
Congestion Control	Enables or disables the congestion control option, which configures Traffic Edge to stop forwarding HTTP requests to origin servers when they become congested and to send the client a message to retry the congested origin server later.
Congestion Rules	Displays the `congestion.config` file so that you can configure congestion control options. Refer to *congestion.config, on page 371*.

ARM

The following table describes the **ARM** configuration options.

The **ARM** configuration options display on the **Configure** tab only if you have enabled ARM.

Option	Description
	General
IP Spoofing	Enables or disables the IP spoofing option, which configures Traffic Edge to establish connections to origin servers with the client IP address instead of the Traffic Edge IP address. HTTP only.
Network Address Translation (NAT)	Displays the redirection rules in the `ipnat.conf` file that specify how incoming packets are readdressed when Traffic Edge is serving traffic transparently. Traffic Edge creates redirection rules during installation. You can modify these rules to suit your needs.
Refresh	Updates the table to display the most up-to-date rules in the `ipnat.conf` file. Click this button after you have added or modified rules with the configuration file editor.

Edit File	Opens the configuration file editor so that you can edit and add rules to the `ipnat.conf` file. The configuration file editor page is described below.
ipnat.conf Configuration File Editor	
rule display box	Lists the `ipnat.conf` file rules. You must select a rule to edit it. The buttons on the left of the box allow you to delete or move the selected rule up or down in the list.
Add	Adds a new rule to the rule display box at the top of the configuration file editor page. You must enter information in the fields provided before you click this button.
Set	Updates the rule display box at the top of the configuration file editor page with any configuration changes you make**.** You must select a rule and change its properties before you click this button.
Ethernet Interface	Specifies the Ethernet interface that traffic uses to access the Traffic Edge machine: for example, `hme0` on Solaris or `eth0` on Linux.
Connection Type	Specifies the connection type that applies for the rule: TCP or UDP.
Source IP	Specifies the IP address from which traffic is sent.
Source CIDR (Optional)	Specifies the IP address in CIDR (Classless Inter-Domain Routing) format, such as 1.1.1.0/24
Source Port	Specifies the traffic destination port: for example, 80 for HTTP traffic.
Destination IP	Specifies the IP address of the Traffic Edge node.
Destination Port	Specifies the Traffic Edge's proxy port: for example, 8080 for HTTP traffic.
User Protocol (Optional)	When `dns` is selected, the ARM redirects DNS traffic to Traffic Edge: otherwise, DNS traffic is bypassed.
Clear Fields	Clears all the fields provided.
Apply	Applies the configuration changes. If you set invalid rules, Traffic Edge displays a warning message when you click the **Apply** button.
Close	Exits the configuration file editor. Important: You must click the **Apply** button before you click **Close**; otherwise, all configuration changes will be lost.
Static Bypass	
Static Bypass	Displays a table listing the rules in the `bypass.config` file that specify static transparency bypass rules. When transparency is enabled, the Traffic Edge uses these rules to determine whether to bypass incoming client requests or attempt to serve them transparently.
Refresh	Updates the table to display the most up-to-date rules in the `bypass.config` file. Click this button after you have added or modified rules with the configuration file editor.
Edit File	Opens the configuration file editor so that you can edit and add rules to the `bypass.config` file. The configuration file editor page is described below.

bypass.config Configuration File Editor	
rule display box	Lists the `bypass.config` file rules. You must select a rule to edit it. The buttons on the left of the box allow you to delete or move the selected rule up or down in the list.
Add	Adds a new rule to the rule display box at the top of the configuration file editor page. You must enter information in the fields provided before you click this button.
Set	Updates the rule display box at the top of the configuration file editor page with any configuration changes you make**.** You must select a rule and change its properties before you click this button.
Rule Type	Specifies the rule type: A **bypass** rule bypasses specified incoming HTTP requests. A **deny_dyn_bypass** rule prevents Traffic Edge from bypassing specified incoming HTTP requests dynamically (a deny bypass rule can prevent Traffic Edge from bypassing itself).
Source IP	Specifies the source IP address in incoming HTTP requests that Traffic Edge must bypass or deny bypass. The IP address can be one of the following: A simple IP address, such as 123.45.67.8 In CIDR (Classless Inter-Domain Routing) format, such as 1.1.1.0/24 A range separated by a dash, such as 1.1.1.1-2.2.2.2 Any combination of the above, separated by commas, such as 1.1.1.0/24, 25.25.25.25, 123.1.23.1-123.1.23.123
Destination IP	Specifies the destination IP address in incoming requests that Traffic Edge must bypass or deny bypass. The IP address can be one of the following: A simple IP address, such as 123.45.67.8 In CIDR (Classless Inter-Domain Routing) format, such as 1.1.1.0/24 A range separated by a dash, such as 1.1.1.1-2.2.2.2 Any combination of the above, separated by commas, such as 1.1.1.0/24, 25.25.25.25, 123.1.23.1-123.1.23.123
Clear Fields	Clears all the fields provided.
Apply	Applies the configuration changes. If you set invalid rules, Traffic Edge displays a warning message when you click the **Apply** button.
Close	Exits the configuration file editor. Important: You must click the **Apply** button before you click **Close**; otherwise, all configuration changes will be lost.
Dynamic Bypass	
Dynamic Bypass	Enables or disables the dynamic bypass option to bypass the Traffic Edge proxy and go directly to the origin server when clients or servers cause problems. Dynamic bypass rules are deleted when you stop Traffic Edge.

Behavior: Non-HTTP, Port 80	Select **Enabled** to enable dynamic bypass when Traffic Edge encounters nonHTTP traffic on port 80. Select **Disabled** to disable dynamic bypass when Traffic Edge encounters nonHTTP traffic on port 80. Select **Source-Destination** to enable dynamic source/ destination bypass when Traffic Edge encounters nonHTTP traffic on port 80. Select **Destination Only** to enable dynamic destination bypass when Traffic Edge encounters nonHTTP traffic on port 80.
Behavior: HTTP 400	Select **Enabled** to enable dynamic bypass when an origin server returns a 400 error. Select **Disabled** to disable dynamic bypass when an origin server returns a 400 error. Select **Source-Destination** to enable dynamic source/ destination bypass when an origin server returns a 400 error. Select **Destination Only** to enable dynamic destination bypass when an origin server returns a 400 error.
Behavior: HTTP 401	Select **Enabled** to enable dynamic bypass when an origin server returns a 401 error. Select **Disabled** to disable dynamic bypass when an origin server returns a 401 error. Select **Source-Destination** to enable dynamic source/ destination bypass when an origin server returns a 401 error. Select **Destination Only** to enable dynamic destination bypass when an origin server returns a 401 error.
Behavior: HTTP 403	Select **Enabled** to enable dynamic bypass when an origin server returns a 403 error. Select **Disabled** to disable dynamic bypass when an origin server returns a 403 error. Select **Source-Destination** to enable dynamic source/ destination bypass when an origin server returns a 403 error. Select **Destination Only** to enable dynamic destination bypass when an origin server returns a 403 error.
Behavior: HTTP 405	Select **Enabled** to enable dynamic bypass when an origin server returns a 405 error. Select **Disabled** to disable dynamic bypass when an origin server returns a 405 error. Select **Source-Destination** to enable dynamic source/ destination bypass when an origin server returns a 405 error. Select **Destination Only** to enable dynamic destination bypass when an origin server returns a 405 error.
Behavior: HTTP 406	Select **Enabled** to enable dynamic bypass when an origin server returns a 406 error. Select **Disabled** to disable dynamic bypass when an origin server returns a 406 error. Select **Source-Destination** to enable dynamic source/ destination bypass when an origin server returns a 406 error. Select **Destination Only** to enable dynamic destination bypass when an origin server returns a 406 error.

Behavior: HTTP 408	Select **Enabled** to enable dynamic bypass when an origin server returns a 408 error. Select **Disabled** to disable dynamic bypass when an origin server returns a 408 error. Select **Source-Destination** to enable dynamic source/ destination bypass when an origin server returns a 408 error. Select **Destination Only** to enable dynamic destination bypass when an origin server returns a 408 error.
Behavior: HTTP 500	Select **Enabled** to enable dynamic bypass when an origin server returns a 500 error. Select **Disabled** to disable dynamic bypass when an origin server returns a 500 error. Select **Source-Destination** to enable dynamic source/ destination bypass when an origin server returns a 500 error. Select **Destination Only** to enable dynamic destination bypass when an origin server returns a 500 error.

WCCP

The following table describes the **WCCP** configuration options.

The **WCCP** configuration options display on the **Configure** tab only if you have enabled WCCP.

If you change any of the **WCCP** options, you must restart Traffic Edge.

Option	Description
	General
WCCP Version	Specifies the version of WCCP running on the router: WCCP v1.0 or WCCP v2.0.
	WCCP v2.0
Security: Enable/Disable	Enables or disables security so that the router and Traffic Edge can authenticate each other. (If you enable security in Traffic Edge, you must also enable security on the router. Refer to your Cisco router documentation.)
Security: Password	Specifies the password that Traffic Edge must use for authentication with the router. This password must match the password on the router.
Configuration	Displays the `wccp_config.xml` file so that you can configure WCCP 2.0 options such as, the IP addresses of your routers, multicast addresses, and service groups. Refer to *wccp_config.xml, on page 460*.
Miscellaneous: Reverse Encapsulation	Enables or disables packet encapsulation mode, which enables Traffic Edge to send encapsulated returned (bypassed) packets to the router.
	WCCP v1.0
WCCP Network Interface	Specifies the Ethernet interface Traffic Edge uses to talk to the WCCP 1.0 -enabled router.
WCCP Router IP Address	Specifies the IP address of the WCCP-enabled router sending traffic to Traffic Edge.

DNS Proxy

The following table describes the **DNS Proxy** configuration options.

The **DNS Proxy** configuration options display on the **Configure** tab only if you have enabled DNS Proxy**.**

Option	Description
DNS Proxy Port	Specifies the port that Traffic Edge uses for DNS traffic. The default port is 53. If you change this option, you must restart Traffic Edge.

DNS Resolver

The following table describes the **DNS Resolver** configuration options.

Option	Description
Resolver	
Local Domain Expansion	Enables or disables local domain expansion so that Traffic Edge can attempt to resolve unqualified hostnames by expanding to the local domain; for example, if a client makes a request to an unqualified host named `hostx` and if the Traffic Edge local domain is `y.com`, the Traffic Edge will expand the hostname to `hostx.y.com`.
Host Database	
DNS Lookup Timeout	Specifies the maximum number of seconds Traffic Edge can wait for a lookup response from the Domain Name Server.
Foreground Timeout	Specifies how long DNS entries can remain in the database before they are flagged as stale; for example, if this timeout is 24 hours and a client requests an entry that has been in the database for 24 hours or longer, Traffic Edge refreshes the entry before serving it. **Caution**: Setting the foreground timeout too low might slow response time. Setting it too high risks accumulation of incorrect information.
Split DNS	
Split DNS	Enables or disables the Split DNS option. When enabled, Traffic Edge can use multiple DNS servers, depending on your security requirements; for example, you can configure Traffic Edge to look to one set of DNS servers to resolve hostnames on your internal network, while allowing DNS servers outside of the firewall to resolve hosts on the Internet. This maintains the security of your intranet, while continuing to provide direct access to sites outside your organization. For information about using Split DNS, refer to *Configuring DNS Server Selection (Split DNS), on page 213*.
Default Domain	Specifies the default domain used for split DNS requests. If a hostname does not include a domain, Traffic Edge appends the default domain name to the hostname before choosing which DNS server to use.
DNS Servers Specification	Displays a table listing the rules in the `splitdns.config` file that control which DNS server Traffic Edge uses for resolving hosts under specific conditions.

Refresh	Updates the table to display the most up-to-date rules in the `splitdns.config` file. Click this button after you have added or modified rules with the configuration file editor.
Edit File	Opens the configuration file editor so that you can edit and add rules to the `splitdns.config` file. The configuration file editor page is described below.
splitdns.config Configuration File Editor	
rule display box	Lists the `splitdns.config` file rules. You must select a rule to edit it. The buttons on the left of the box allow you to delete or move the selected rule up or down in the list.
Add	Adds a new rule to the rule display box at the top of the configuration file editor page. You must enter information in the fields provided before you click this button.
Set	Updates the rule display box at the top of the configuration file editor page with any configuration changes you make**.** You must select a rule and change its properties before you click this button.
Primary Destination Type	Specifies that DNS server selection is based on the destination domain (dest_domain), destination host (dest_host), or on a regular expression (url_regex).
Primary Destination Value	Specifies the value of the primary destination.
DNS Server IP	Specifies the DNS server that Traffic Edge must use with the given destination specifier. You can specify a port using a colon (:). If you do not specify a port, 53 is used. You can specify multiple DNS servers separated by spaces or by semicolons (;).
Default Domain Name (Optional)	Specifies the default domain name to use for resolving hosts. Only one entry is allowed. If you do not provide the default domain, the system determines its value from `/etc/resolv.conf` in UNIX or from the Registry in Windows.
Domain Search List (Optional)	Specifies the domain search order. If you do not provide the search list, the system determines the value from `/etc/resolv.conf` in UNIX or from the Registry in Windows.
Clear Fields	Clears all the fields provided.
Apply	Applies the configuration changes. If you set invalid rules, Traffic Edge displays a warning message when you click the **Apply** button.
Close	Exits the configuration file editor. Important: You must click the **Apply** button before you click **Close**; otherwise, all configuration changes will be lost.

Virtual IP

The following table describes the **Virtual IP** configuration options.

Note The **Virtual IP** configuration options display on the **Configure** tab only if you have enabled Virtual IP.

Option	Description
Virtual IP Addresses	Displays a table listing the virtual IP addresses that Traffic Edge manages.
Refresh	Updates the table to display the most up-to-date list of virtual IP addresses. Click this button after you have added to or modified the list of virtual IP addresses with the configuration file editor.
Edit File	Opens the configuration file editor so that you can edit and add to the list of virtual IP addresses. The configuration file editor page is described below.
vaddrs.config Configuration File Editor	
rule display box	Lists the virtual IP addresses. You must select a virtual IP address to edit it. The buttons on the left of the box allow you to delete or move the selected virtual IP address up or down in the list.
Add	Adds a new virtual IP address to the rule display box at the top of the configuration file editor page. You must enter information in the fields provided before you click this button.
Set	Updates the rule display box at the top of the configuration file editor page with any configuration changes you make**.** You must select a virtual IP address and change its properties before you click this button.
Virtual IP Address	Specifies the virtual IP address managed by Traffic Edge.
Ethernet Interface	Specifies the network interface assigned to the virtual IP address.
Sub-Interface	Specifies the subinterface ID. This is a number between 1 and 255 that the interface uses for the address.
Clear Fields	Clears all the fields provided.
Apply	Applies the configuration changes. If you set invalid rules, Traffic Edge displays a warning message when you click the **Apply** button.
Close	Exits the configuration file editor. Important: You must click the **Apply** button before you click **Close**; otherwise, all configuration changes will be lost.

Plugins

The **Plugins** section lists the plugins currently running on your Traffic Edge that you can configure using Traffic Manager. A plugin is a program that extends the functionality of Traffic Edge; for example, you can use specific plugins to blacklist origin servers, filter web content, and authenticate users.

To configure a plugin, click the plugin in the list. A configuration page for the plugin opens.

If no plugins configurable from Traffic Manager are running on your Traffic Edge node, the message `No currently installed plugins` displays on the **Plugins** page.

If you are interested in developing a plugin, refer to the Traffic Edge Software Development Kit (SDK) at `http://www.inktomi.com/products/network/partners/`.

APPENDIX C

Traffic Line Commands

This appendix contains the following sections:

Traffic Line Commands

Use Traffic Line to execute individual Traffic Edge commands and to script multiple commands in a shell.

You execute Traffic Line commands from the Traffic Edge `bin` directory. (In Windows, use a command prompt window.)

In UNIX, if the Traffic Edge `bin` directory is not in your path, prepend the Traffic Line command with `./` (for example, `./traffic_line -p`).

The following table describes all the commands available in Traffic Line.

Command	Description
`traffic_line -b`	Bounces the Traffic Edge on the local node. Bouncing the Traffic Edge shuts down and immediately restarts the Traffic Edge node.
`traffic_line -c`	Clears the accumulated statistics on the local node.
`traffic_line -h`	Displays the list of Traffic Line commands.
`traffic_line -p` *`socket_path`*	Specifies the location (directory and path) of the socket used for Traffic Line and Traffic Manager communication. The default path is: *`install_dir`*`/config/cli`
`traffic_line -q`	Displays a list of the origin servers that are currently congested. Note: To use this command, you must set the variable `proxy.config.raf.enabled` to 1 and then set the variable `proxy.config.raf.port` to a different port *only* if there is a conflict with the default port 9000.
`traffic_line -r` *`variable`*	Displays specific performance statistics or a current configuration setting. For a list of the variables you can specify, refer to *Traffic Line Variables, on page 353*.

Command	Description
`traffic_line -s variable -v value`	Sets configuration variables. `variable` is the configuration variable you want to change and `value` is the value you want to set. Refer to *records.config, on page 397*, for a list of the configuration variables you can specify.
`traffic_line -x`	Initiates a Traffic Edge configuration file reread. Executing this command is similar to clicking the **Apply** button in Traffic Manager. Use this command after every configuration file modification.
`traffic_line -B`	Bounces all the Traffic Edge nodes in the cluster. Bouncing the Traffic Edge shuts down and immediately restarts Traffic Edge node by node.
`traffic_line -C`	Clears the accumulated statistics on all the nodes in the cluster.
`traffic_line -L`	Restarts the `traffic_manager` process and the `traffic_server` process on the local node.
`traffic_line -M`	Restarts the `traffic_manager` process and the `traffic_server` process on all the nodes in a cluster.
`traffic_line -S`	Shuts down the Traffic Edge on the local node.
`traffic_line -U`	Starts the Traffic Edge on the local node.

Traffic Line Variables

You can view statistics and change configuration options in Traffic Line by using specific variables. The variables used for gathering statistics are described below.

The variables used for viewing and changing configuration options are described in *records.config, on page 397*. For procedures on how to specify the variables, refer to *Viewing Statistics from Traffic Line, on page 188*, and *Configuring Traffic Edge Using Traffic Line, on page 196*.

The variables used for viewing individual statistics are described in the following table. For a detailed description of the statistics, refer to *Appendix A, Traffic Manager Statistics*. To view a statistic in Traffic Line, enter the command `traffic_line -r variable` at the prompt.

Statistic	Variable
Summary	
Node name	`proxy.node.hostname`
Objects served	`proxy.node.user_agents_total_documents_served`
Transactions per second	`proxy.node.user_agent_xacts_per_second`
Node	
Document hit rate	`proxy.node.cache_hit_ratio_avg_10s` `proxy.cluster.cache_hit_ratio_avg_10s`
Bandwidth savings	`proxy.node.bandwidth_hit_ratio_avg_10s` `proxy.cluster.bandwidth_hit_ratio_avg_10s`
Cache percent free	`proxy.node.cache.percent_free` `proxy.cluster.cache.percent_free`
Open origin server connections	`proxy.node.current_server_connections` `proxy.cluster.current_server_connections`
Open client connections	`proxy.node.current_client_connections` `proxy.cluster.current_client_connections`
Cache transfers in progress	`proxy.node.current_cache_connections` `proxy.cluster.current_cache_connections`
Client throughput (Mbits/sec)	`proxy.node.client_throughput_out` `proxy.cluster.client_throughput_out`
Transactions per second	`proxy.node.user_agent_xacts_per_second` `proxy.cluster.user_agent_xacts_per_second`
DNS lookups per second	`proxy.node.dns.lookups_per_second` `proxy.cluster.dns.lookups_per_second`
Host database hit rate	`proxy.node.hostdb.hit_ratio_avg_10s` `proxy.cluster.hostdb.hit_ratio_avg_10s`
HTTP	
Total document bytes from client	`proxy.process.http.user_agent_response_document_total_size`
Total header bytes from client	`proxy.process.http.user_agent_response_header_total_size`
Total connections to client	`proxy.process.http.total_client_connections`
Client transactions in progress	`proxy.process.http.current_client_transactions`

Total document bytes from origin server	proxy.process.http.origin_server_response_document_total_size
Total header bytes from origin server	proxy.process.http.origin_server_response_header_total_size
Total connections to origin server	proxy.process.http.total_server_connections
Origin server transactions in progress	proxy.process.http.current_server_transactions
FTP	
Currently open FTP connections	proxy.process.ftp.connections_currently_open
Successful PASV connections	proxy.process.ftp.connections_successful_pasv
Unsuccessful PASV connections	proxy.process.ftp.connections_failed_pasv
Successful PORT connections	proxy.process.ftp.connections_successful_port
Unsuccessful PORT connections	proxy.process.ftp.connections_failed_port
ICP	
ICP query requests originating from this node	proxy.process.icp.icp_query_requests
ICP query messages sent from this node	proxy.process.icp.total_udp_send_queries
ICP peer hit messages received from this node	proxy.process.icp.icp_query_hits
ICP peer miss messages received from this node	proxy.process.icp.icp_query_misses
Total ICP responses received from this node	proxy.process.icp.icp_remote_responses
Average ICP message response time (ms) from this node	proxy.process.icp.total_icp_response_time
Average ICP request time (ms) from this node	proxy.process.icp.total_icp_request_time
Query messages received from ICP peers	proxy.process.icp.icp_remote_query_requests
Remote query hits from ICP peers	proxy.process.icp.cache_lookup_success
Remote query misses from ICP peers	proxy.process.icp.cache_lookup_fail
Successful response messages sent to peers	proxy.process.icp.query_response_write
Real Networks	
Open client on-demand connections	proxy.process.rni.current_client_connections
Client on-demand requests	proxy.process.rni.downstream_requests
Client on-demand response bytes	proxy.process.rni.downstream.response_bytes

Open client live connections	proxy.process.rni.current_live_streams
Number of client live requests	proxy.process.rni.total_live_streams
Server on-demand response bytes	proxy.process.rni.upstream.response_bytes
Total bytes hit from the cache	proxy.process.rni.byte_hit_sum
Total bytes missed from the cache	proxy.process.rni.byte_miss_sum
QuickTime	
Open client connections	proxy.process.qt.current_client_connections
Client on-demand requests	proxy.process.qt.downstream_requests
Client on-demand response bytes	proxy.process.qt.downstream.response_bytes
Server on-demand requests	proxy.process.qt.upstream_requests
Server on-demand response bytes	proxy.process.qt.upstream.response_bytes
Total bytes hit from the cache	proxy.process.qt.byte_hit_sum
Total Bytes Missed from the Cache	proxy.process.qt.byte_miss_sum
WMT	
Open client connections	proxy.process.wmt.current_client_connections
Client on-demand requests	proxy.process.wmt.downstream_requests
Client on-demand response bytes	proxy.process.wmt.downstream.response_bytes
Server on-demand requests	proxy.process.wmt.upstream_requests
Server on-demand response bytes	proxy.process.wmt.upstream.response_bytes
Total bytes hit from the cache	proxy.process.wmt.byte_hit_sum
Total Bytes Missed from the Cache	proxy.process.wmt.byte_miss_sum
WCCP	
WCCP router's IP address	proxy.node.wccp.router_ip
WCCP router status	proxy.node.wccp.router_status
WCCP node IP address	proxy.node.wccp.my_ip
Percentage of WCCP traffic received	proxy.node.wccp.my_share
Number of WCCP heartbeats	proxy.node.wccp.hbeats_received
WCCP leader's IP address	proxy.node.wccp.leader_ip
Number of active WCCP nodes	proxy.node.wccp.number_of_caches_up
Cache	
Bytes used	proxy.process.cache.bytes_used
Cache size	proxy.process.cache.bytes_total
Lookups in progress	proxy.process.cache.lookup.active
Lookups completed	proxy.process.cache.lookup.success
Lookup misses	proxy.process.cache.lookup.failure

Reads in progress	proxy.process.cache.read.active
Reads completed	proxy.process.cache.read.success
Read misses	proxy.process.cache.read.failure
Writes in progress	proxy.process.cache.write.active
Writes completed	proxy.process.cache.write.success
Write failures	proxy.process.cache.write.failure
Updates in progress	proxy.process.cache.update.active
Updates completed	proxy.process.cache.update.success
Update failures	proxy.process.cache.update.failure
Removes in progress	proxy.process.cache.remove.active
Remove successes	proxy.process.cache.remove.success
Remove failures	proxy.process.cache.remove.failure
Host Database	
Total lookups	proxy.process.hostdb.total_lookups
Total hits	proxy.process.hostdb.total_hits
Time TTL (min)	proxy.process.hostdb.ttl
DNS	
DNS total lookups	proxy.process.dns.total_dns_lookups
Average lookup time (ms)	proxy.process.dns.lookup_avg_time
DNS successes	proxy.process.dns.lookup_successes
Cluster	
Bytes read	proxy.process.cluster.read_bytes
Bytes written	proxy.process.cluster.write_bytes
Connections open	proxy.process.cluster.connections_open
Total operations	proxy.process.cluster.connections_opened
Network backups	proxy.process.cluster.net_backup
Clustering nodes	proxy.process.cluster.nodes
SOCKS	
Unsuccessful connections	proxy.process.socks.connections_unsuccessful
Successful connections	proxy.process.socks.connections_successful
Connections in progress	proxy.process.socks.connections_currently_open
Logging	
Currently open log files	proxy.process.log2.log_files_open
Space used for log files	proxy.process.log2.log_files_space_used
Number of access events logged	proxy.process.log2.event_log_access
Number of access events skipped	proxy.process.log2.event_log_access_skip
Number of error events logged	proxy.process.log2.event_log_error

Congestion Control	
Number of congestions Traffic Edge has observed because the maximum number of connections were exceeded.	`proxy.process.congestion.congested_on_max_connection`
Number of congestions Traffic Edge has observed because of an operating system response or timeout failure.	`proxy.process.congestion.congested_on_conn_failures`

APPENDIX D

Event Logging Formats

This appendix contains the following sections:

- *Inktomi Custom Logging Fields*, below, provides descriptions of Inktomi logging fields.
- *Logging Format Cross-Reference, on page 363*, provides cross-references between Inktomi logging fields and Netscape and Squid logging fields (including Netscape Extended and Extended-2 fields).

Inktomi Custom Logging Fields

The following table describes Inktomi custom logging fields.

%<field symbol>	Description
{HTTP header field name}cqh	Logs the information in the requested field of the client request HTTP header; for example, `%<{Accept-Language}cqh>` logs the Accept-Language: field in client request headers.
{HTTP header field name}pqh	Logs the information in the requested field of the proxy request HTTP header; for example, `%<{Authorization}pqh>` logs the Authorization: field in proxy request headers.
{HTTP header field name}psh	Logs the information in the requested field of the proxy response HTTP header; for example, `%<{Retry-After}psh>` logs the Retry-After: field in proxy response headers.
{HTTP header field name}ssh	Logs the information in the requested field of the server response HTTP header; for example, `%<{Age}ssh>` logs the Age: field in server response headers.
band	The bandwidth of data sent to the client player, in bits per second for WMT streams. The value for the band field is computed in the same way for QuickTime and WMT.
caun	The client authenticated username; result of the RFC931/ident lookup of the client username.
cfsc	The client finish status code; specifies whether the client request to Traffic Edge was successfully completed (FIN) or interrupted (INTR).
cgid	The client GUID (global unique identifier): a special string that the client sends to the RealServer indicating that it is registering a request.
chi	The IP address of the client's host machine. For streaming media, if the request is from a RealOne player and the RTSP proxy port is set to the default port 554 either explicitly or transparently, this field shows the IP address of Traffic Edge instead of the IP address of the client. For streaming media PNA requests, the value for this field is always 0.

%<field symbol>	Description
cqbl	The client request transfer length; the body length in the client request to Traffic Edge in bytes.
cqhl	The client request header length; the header length in the client request to Traffic Edge.
cqhm	The HTTP method in the client request to Traffic Edge: GET, POST, and so on (subset of cqtx).
cqhv	The client request HTTP version.
cqtd	The client request timestamp; specifies the date of the client request in the format yyyy-mm-dd, where yyyy is the 4-digit year, mm is the 2-digit month, and dd is the 2-digit day.
cqtn	The client request timestamp; date and time of the client's request (in the Netscape timestamp format).
cqtq	The client request timestamp with millisecond resolution.
cqts	The client-request timestamp in Squid format; the time of the client request in seconds since January 1, 1970 UTC (with millisecond resolution).
cqtt	The client request timestamp; the time of the client request in the format hh:mm:ss, where hh is the two-digit hour in 24-hour format, mm is the two-digit minutes, and ss is the 2-digit seconds; for example, 16:01:19.
cqtx	The full HTTP client request text, minus headers; for example, `GET http://www.company.com HTTP/1.0` In reverse proxy mode, Traffic Edge logs the rewritten (mapped) URL (according to the rules in the `remap.config` file), *not* the pristine (unmapped) URL. To configure Traffic Edge to log the original, unmapped URL, set the variable `proxy.config.url_remap.pristine_host_hdr` in the `records.config` file to `1`.
cqu	The universal resource identifier (URI) of the request from client to Traffic Edge (subset of `cqtx`). In reverse proxy mode, Traffic Edge logs the rewritten (mapped) URL (according to the rules in the `remap.config` file), *not* the pristine (unmapped) URL. To configure Traffic Edge to log the original, unmapped URL, set the variable `proxy.config.url_remap.pristine_host_hdr` in the `records.config` file to `1`.
cquc	The client request canonical URL; differs from cqu in that blanks (and other characters that might not be parsed by log analysis tools) are replaced by escape sequences. The escape sequence is a percentage sign followed by the ASCII code number in hex. In reverse proxy mode, Traffic Edge logs the rewritten (mapped) URL (according to the rules in the `remap.config` file), *not* the pristine (unmapped) URL. To configure Traffic Edge to log the original, unmapped URL, set the variable `proxy.config.url_remap.pristine_host_hdr` in the `records.config` file to `1`.

%<field symbol>	Description
cqup	The client request URL path; specifies the argument portion of the URL (everything after the host); for example, if the URL is `http://www.company.com/images/x.gif`, then this field displays `/images/x.gif`.
cqus	The client request URL scheme (HTTP, FTP, etc.).
cquuc	The client request unmapped URL canonical. This field records a URL before it is remapped (reverse proxy mode).
crc	The cache result code; specifies how the cache responded to the request (HIT, MISS, and so on).
fsiz	The size of the file, in bytes, as seen by the origin server. Windows Media Player sees a smaller size in the case of multibitrate clips.
pfsc	The proxy finish status code; specifies whether the Traffic Edge request to the origin server was successfully completed (FIN) or interrupted (INTR).
phn	The hostname of the Traffic Edge that generated the log entry in collated log files.
phr	The proxy hierarchy route; the route that Traffic Edge used to retrieve the object. Refer to *phr, on page 274* for information about streaming media transactions.
pqbl	The proxy request transfer length; the body length in the Traffic Edge request to the origin server.
pqhl	The proxy request header length; the header length in the Traffic Edge request to the origin server.
pqsi	The proxy request server IP address (0 on cache hits and parent-ip for requests to parent proxies).
pqsn	The proxy request server name; the name of the server that fulfilled the request.
prcb	The number of proxy response bytes to the client from the cache. This value is always zero for live streams.
prob	The number of proxy response bytes to the client from the origin server. When Traffic Edge serves a WMT stream from the cache, a small number of bytes are logged as coming from the origin server. If the client uses Windows Media Player Version 6, more bytes are logged than if the client is using Windows Media Player Version 7. This is the result of Windows Media Player behavior. For QuickTime, the `prob` field is always a nonzero number on cache hits. Traffic Edge tracks both the control and data bytes obtained from the QuickTime origin server. The control bytes are the product of RTSP control commands such as `DESCRIBE`, `SETUP`, and `PLAY` and are typically 2 to 3 kilobytes per transaction. These values result in a nonzero value for the `prob` field even when streaming media data is served from the cache.
pscl	The length of the Traffic Edge response to the client in bytes.
psct	The content type of the document from server response header: for example, `img/gif`.
pshl	The header length in the Traffic Edge response to the client.

%<field symbol>	Description
psql	The proxy response transfer length in Squid format (includes header and content length).
pssc	The HTTP response status code from Traffic Edge to the client.
shi	The IP address resolved from the DNS name lookup of the host in the request. For hosts with multiple IP addresses, this field records the IP address resolved from that particular DNS lookup. This can be misleading for cached documents; for example, if the first request was a cache miss and came from IP1 for server S and the second request for server S resolved to IP2 but came from the cache, the log entry for the second request will show IP2.
shn	The hostname of the origin server.
sscl	The response length, in bytes, from origin server to Traffic Edge.
sshl	The header length in the origin server response to Traffic Edge, in bytes.
sshv	The server response HTTP version (1.0, 1.1, and so on).
sssc	The HTTP response status code from origin server to Traffic Edge.
styp	Indicates if the stream was live or on-demand and if it was authenticated and provides information about the circumstances under which the stream was or was not cached. Refer to *styp, on page 275*.
tfcb	The time to first client byte, in milliseconds. For quality of service logging. Streaming media only.
ttms	The time Traffic Edge spends processing the client request; the number of milliseconds between the time that the client establishes the connection with Traffic Edge and the time that Traffic Edge sends the last byte of the response back to the client.
ttmsf	The time Traffic Edge spends processing the client request as a fractional number of seconds; specifies the time in millisecond resolution, but instead of formatting the output as an integer (as with ttms), the display is formatted as a floating-point number representing a fractional number of seconds; for example, if the time is 1500 milliseconds, this field displays 1.5 while the ttms field displays 1500 and the tts field displays 1.
tts	The time Traffic Edge spends processing the client request; the number of seconds between the time that the client establishes the connection with Traffic Edge and the time that Traffic Edge sends the last byte of the response back to the client.

Logging Format Cross-Reference

The following sections illustrate the correspondence between Inktomi logging fields and standard logging fields for the Squid and Netscape formats.

Squid Logging Formats

The following table lists the Squid logging fields and the corresponding Inktomi logging field symbols.

Squid	Inktomi Field Symbols
time	cqts
elapsed	ttms
client	chi
action/code	crc/pssc
size	psql
method	cqhm
url	cquc
ident	caun
hierarchy/from	phr/pqsn
content	psct

Netscape Common Logging Formats

The following table lists the Netscape Common logging fields and the corresponding Inktomi logging field symbols.

Netscape Common	Inktomi Field Symbols
host	chi
usr	caun
[time]	[cqtn]
“req”	“cqtx”
s1	pssc
c1	pscl

Netscape Extended Logging Formats

The following table lists the Netscape Extended logging fields and the corresponding Inktomi logging field symbols.

Netscape Extended	Inktomi Field Symbols
host	chi
usr	caun
[time]	[cqtn]
“req”	“cqtx”
s1	pssc

Netscape Extended	Inktomi Field Symbols
c1	pscl
s2	sssc
c2	sscl
b1	cqbl
b2	pqbl
h1	cqhl
h2	pshl
h3	pqhl
h4	sshl
xt	tts

Netscape Extended-2 Logging Formats

The following table lists the Netscape Extended-2 logging fields and the corresponding Inktomi logging field symbols.

Netscape Extended-2	Inktomi Field Symbols
host	chi
usr	caun
[time]	[cqtn]
“req”	“cqtx”
s1	pssc
c1	pscl
s2	sssc
c2	sscl
b1	cqbl
b2	pqbl
h1	cqhl
h2	pshl
h3	pqhl
h4	sshl
xt	tts
route	phr
pfs	cfsc
ss	pfsc
crc	crc

APPENDIX E Configuration Files

This appendix describes the Traffic Edge configuration files that you can edit. For a list of all the configuration files used by Traffic Edge, including files that you must not edit, refer to the *Traffic Edge Installation Guide*.

- *winnt_intr.config, on page 463*
- *Specifying URL Regular Expressions (url_regex), on page 464*

arm_security.config

The `arm_security.config` file defines which hosts are allowed to communicate with the Traffic Edge machine using TCP and UDP through defined ports. Traffic Edge uses this configuration file when the ARM security option is enabled. For information about enabling the ARM security option, refer to *Controlling Host Access to the Traffic Edge Machine, on page 204*.

In the `arm_security.config` file you can specify the following information:

- The ports that are open by default, for either TCP or UDP
- The hosts that are denied access to specific destination ports, for either TCP or UDP
- The hosts that are allowed access to specific destination ports, for either TCP or UDP

CAUTION Before you enable the ARM security option, ensure that you have either console access to the Traffic Edge machine or that you have added the appropriate rules to the configuration file to allow telnet or ssh access for yourself.

The ports you specify in the access control list remain closed even when Traffic Edge is not running.

IMPORTANT After you modify the `arm_security.config` file, you must restart Traffic Edge.

Format

Each line in the `arm_security.config` file uses one of the following formats:

`open tcp|udp ports` *`o_ports`*

`deny tcp|udp dport` *`d_ports`* `src` *`src_ipaddress`*

`allow tcp|udp src` *`src_ipaddress`* `dst` *`dst_ipaddress`* `dport` *`d_ports`* `sport|`*`s_ports`*

The following table describes each field.

Field	Allowed Inputs
`o_ports`	The specific port or range of ports that are open by default.
`d_ports`	The destination port, or range of destination ports, through which TCP traffic is either allowed or denied. To specify a series of ports, you must use the parameter `dports` instead of `dport` and separate each port number with a space: for example, `deny tcp dports 2121 2130 2140`.
`s_ports`	The source port, or range of source ports, from which TCP traffic is allowed. To specify a series of ports, you must use the parameter `sports` instead of `sport` and separate each port number with a space: for example, `allow tcp src 11.11.1.1 sports 2121 2130 2140`.

Field	Allowed Inputs
src_ipaddress	The IP address, or range of IP addresses, specifying the source of the communication.
dst_ipaddress	The IP address, or range of IP addresses, specifying the destination of the communication.

Note The `arm_security.config` file does not support spaces before or after the hyphen in an IP address range; for example, you must specify `12.34.56.1-12.34.56.7`.

Examples

The following example defines port 80 as open for TCP communication. All other ports are closed.

```
open tcp ports 80
```

In the following example, the first line specifies that all hosts are denied access to destination port 80 using TCP. The second line specifies that host 209.1.2.2 is denied access to destination port 90 using UDP.

```
deny tcp dport 80 src 0.0.0.0-255.255.255.255
deny udp dport 90 src 209.1.2.2
```

In the following example, the first line specifies that host 111.11.11.1 using source port 20 is allowed to communicate with host 123.45.67.8 on destination ports 127-130 using TCP. The second line specifies that all hosts are allowed to communicate with host 123.12.3.4 using UDP.

```
allow tcp src 111.11.11.1 dst 123.45.67.8 dport 127-130 sport 20
allow udp dst 123.12.3.4
```

bypass.config

The `bypass.config` file contains static bypass rules that Traffic Edge uses in transparent proxy caching mode. *Static bypass* rules instruct Traffic Edge to bypass certain incoming client requests so that they are served by the origin server.

The `bypass.config` file also accepts *dynamic* deny bypass rules; refer to *Dynamic Deny Bypass Rules, on page 368*.

You can configure three types of static bypass rules:

- *Source bypass* rules configure Traffic Edge to bypass a particular source IP address or range of IP addresses; for example, you can bypass clients that do not want to use caching.
- *Destination bypass* rules configure Traffic Edge to bypass a particular destination IP address or range of IP addresses; for example, you can bypass origin servers that use IP authentication based on the client's real IP address.

IMPORTANT Destination bypass rules prevent Traffic Edge from caching an entire site. You will experience hit rate impacts if the site you bypass is popular.

- *Source/destination pair bypass* rules configure Traffic Edge to bypass requests that originate from the specified source to the specified destination; for example, you can route around specific client-server pairs that experience broken IP authentication or out-of-band HTTP traffic problems when cached. Source/destination bypass rules can be preferable to destination rules because they block a destination server only for users that experience problems.

IMPORTANT After you modify the `bypass.config` file, you must restart Traffic Edge.

Format

Bypass rules have the following format:

`bypass src` *`ipaddress`* `|` `dst` *`ipaddress`* `|` `src` *`ipaddress`* `AND dst` *`ipaddress`*

The following table describes the options.

Option	Description
`src` *`ipaddress`*	Specifies the source (client) IP address in incoming requests that Traffic Edge must bypass. *`ipaddress`* can be one of the following: A simple IP address, such as 123.45.67.8 In CIDR (Classless Inter-Domain Routing) format, such as 1.1.1.0/24 A range separated by a dash, such as 1.1.1.1-2.2.2.2 Any combination of the above, separated by commas, such as 1.1.1.0/24, 25.25.25.25, 123.1.23.1-123.1.23.123
`dst` *`ipaddress`*	Specifies the destination (origin server) IP address in incoming requests that Traffic Edge must bypass. *`ipaddress`* can be one of the following: A simple IP address, such as 123.45.67.8 In CIDR (Classless Inter-Domain Routing) format, such as 1.1.1.0/24 A range separated by a dash, such as 1.1.1.1-2.2.2.2 Any combination of the above, separated by commas, such as 1.1.1.0/24, 25.25.25.25, 123.1.23.1-123.1.23.123
`src` *`ipaddress`* `AND dst` *`ipaddress`*	Specifies the source and destination IP address pair that Traffic Edge must bypass. *`ipaddress`* must be a single IP address, such as 123.45.67.8

Dynamic Deny Bypass Rules

In addition to static bypass rules, the `bypass.config` file also accepts *dynamic deny* bypass rules. Deny bypass rules prevent Traffic Edge from bypassing certain incoming client requests dynamically (a deny bypass rule can prevent Traffic Edge from bypassing itself). Dynamic deny bypass rules can be source, destination, or source/destination and have the following format:

`deny_dyn_bypass src` *`ipaddress`* `|` `dst` *`ipaddress`* `|` `src` *`ipaddress`* `AND dst` *`ipaddress`*

For a description of the options, refer to the table above.

For the dynamic deny bypass rules to work, you must enable the **Dynamic Bypass** option in Traffic Manager or set the variable `proxy.config.arm.bypass_dynamic_enabled` to `1` in the `records.config` file.

IMPORTANT Static bypass rules overwrite dynamic deny bypass rules. Therefore, if a static bypass rule and a dynamic bypass rule contain the same IP address, the dynamic deny bypass rule is ignored.

Examples

The following example shows source, destination, and source/destination *bypass* rules:

```
bypass src 1.1.1.0/24, 25.25.25.25, 128.252.11.11-128.252.11.255
bypass dst 24.24.24.0/24
bypass src 25.25.25.25 AND dst 24.24.24.0
```

The following example shows source, destination, and source/destination *dynamic deny bypass* rules:

```
deny_dyn_bypass src 128.252.11.11-128.252.11.255
deny_dyn_bypass dst 111.111.11.1
deny_dyn_bypass src 111.11.11.1 AND dst 111.11.1.1
```

cache.config

The `cache.config` file defines how Traffic Edge caches web objects. You can add caching rules to specify the following:

- Not to cache objects from specific IP addresses
- How long to pin particular objects in the cache
- How long to consider cached objects as fresh
- Whether to ignore no-cache directives from the server

IMPORTANT After you modify the `cache.config` file, navigate to the Traffic Edge `bin` directory (in Windows, open a Command Prompt window and navigate to the `Traffic Edge bin` directory) and then run the `traffic_line -x` command to apply the changes. When you apply the changes to a node in a cluster, Traffic Edge automatically applies the changes to all other nodes in the cluster.

Format

Each line in the `cache.config` file contains a caching rule. Traffic Edge recognizes three space-delimited tags:

```
primary_destination=value secondary_specifier=value action=value
```

You can use more than one secondary specifier in a rule. However, you cannot repeat a secondary specifier.

The following table lists the possible primary destinations and their allowed values.

Primary Destination	Allowed Value
dest_domain	A requested domain name. Traffic Edge matches the domain name of the destination from the URL in the request.
dest_host	A requested hostname. Traffic Edge matches the hostname of the destination from the URL in the request.
dest_ip	A requested IP address. Traffic Edge matches the IP address of the destination in the request.
url_regex	A regular expression to be found in a URL.

The secondary specifiers are optional in the cache.config file. The following table lists the possible secondary specifiers and their allowed values.

Secondary Specifier	Allowed Value
port	A requested URL port.
scheme	A request URL protocol; http, https, ftp, rtsp, or mms.
prefix	A prefix in the path part of a URL.
suffix	A file suffix in the URL.
method	A request URL method: get, put, post, trace.
time	A time range, such as 08:00-14:00.
src_ip	A client IP address.
tag	Specifies Real Networks or QuickTime streams. You can enter RNI or QT: for example, tag=QT.

The following table lists the possible actions and their allowed values.

Action	Value
action	One of the following values: never-cache configures Traffic Edge to never cache specified objects. ignore-no-cache configures Traffic Edge to ignore all Cache-Control: no-cache headers. ignore-client-no-cache configures Traffic Edge to ignore Cache-Control: no-cache headers from client requests. ignore-server-no-cache configures Traffic Edge to ignore Cache-Control: no-cache headers from origin server responses.
pin-in-cache	The amount of time you want to keep the object(s) in the cache. The following time formats are allowed: d for days: for example 2d h for hours: for example, 10h m for minutes: for example, 5m s for seconds: for example, 20s mixed units: for example, 1h15m20s

Action	Value
revalidate	The amount of time you want to consider the object(s) fresh. Use the same time formats as pin-in-cache.
ttl-in-cache	The amount of time you want to keep objects in the cache regardless of Cache-Control response headers. Use the same time formats as pin-in-cache and revalidate.

Examples

The following example configures Traffic Edge to never cache FTP documents requested from the IP address 112.12.12.12:

```
dest_ip=112.12.12.12 scheme=ftp action=never-cache
```

The following example configures Traffic Edge to keep documents with URLs that contain the regular expression politics and the path prefix /viewpoint in the cache for 12 hours:

```
url_regex=politics prefix=/viewpoint pin-in-cache=12h
```

The following example configures Traffic Edge to revalidate gif and jpeg objects in the domain mydomain.com every 6 hours and all other objects in mydomain.com every hour:

```
dest_domain=mydomain.com suffix=gif revalidate=6h
dest_domain=mydomain.com suffix=jpeg revalidate=6h
dest_domain=mydomain.com revalidate=1h
```

The rules are applied in the order listed.

congestion.config

The congestion.control file lets you configure Traffic Edge to stop forwarding HTTP requests to origin servers when they become congested and to send the client a message to retry the congested origin server later.

You can create rules in the congestion.config file to specify:

- Which origin servers Traffic Edge tracks for congestion
- The timeouts Traffic Edge uses, depending on whether a server is congested
- The page that Traffic Edge sends to the client when a server becomes congested
- If Traffic Edge tracks the origin servers per IP address or per hostname
- If Traffic Edge sends SNMP traps when a server becomes congested

IMPORTANT After you modify the congestion.control file, navigate to the Traffic Edge bin directory (in Windows, open a Command Prompt window and navigate to the Traffic Edge bin directory) and then run the traffic_line -x command to apply the changes. When you apply the changes to a node in a cluster, Traffic Edge automatically applies the changes to all other nodes in the cluster.

Traffic Edge uses the congestion.config file only if you enable the congestion control option; refer to *Using Congestion Control, on page 58*.

Format

Each line in the `congestion.config` file must follow the following format. Traffic Edge applies the rules in the order listed, starting at the top of the file.

Traffic Edge recognizes three space-delimited tags:

`primary_destination=value secondary_specifier=value action=value`

The following table lists the possible primary destinations and their allowed values.

Primary Destination	Allowed Value
`dest_domain`	A requested domain name.
`dest_host`	A requested hostname.
`dest_ip`	A requested IP address.
`url_regex`	A regular expression to be found in a URL.

The secondary specifiers are optional in the `congestion.config` file. The following table lists the possible secondary specifiers and their allowed values.

Note You can use more than one secondary specifier in a rule. However, you cannot repeat a secondary specifier.

Secondary Specifier	Allowed Value
`port`	A requested URL port or range of ports.
`prefix`	A prefix in the path part of a URL.

The following table lists the possible tags and their allowed values.

Tag	Allowed Value
`max_connection_failures`	Specifies the maximum number of connection failures allowed within the fail window described below before Traffic Edge marks the origin server as congested. The default value is 5.
`fail_window`	Specifies the time period during which the maximum number of connection failures can occur before Traffic Edge marks the origin server as congested. The default value is 120 seconds.
`proxy_retry_interval`	Specifies the number of seconds that Traffic Edge waits before contacting a congested origin server again. The default value is 10 seconds
`client_wait_interval`	Specifies the number of seconds that the client is advised to wait before retrying the congested origin server. The default value is 300 seconds.
`wait_interval_alpha`	Specifies the upper limit for a random number, which is added to the wait interval. The default value is 30 seconds.
`live_os_conn_timeout`	Specifies the connection timeout to the live (uncongested) origin server. The default value is 60 seconds. If a client stops a request before the timeout occurs, Traffic Edge does not record a connection failure.
`live_os_conn_retries`	Specifies the maximum number of retries allowed to the live (uncongested) origin server. The default value is 2.

Tag	Allowed Value
dead_os_conn_timeout	Specifies the connection timeout to the congested origin server. The default value is 15 seconds.
dead_os_conn_retries	Specifies the maximum number of retries allowed to the congested origin server. The default value is 1.
max_connection	Specifies the maximum number of connections allowed from Traffic Edge to the origin server. The default value is -1.
error_page	Specifies the error page that is sent to the client when a server is congested. You must enclose the value in quotes. The default value is "congestion#retryAfter". The error page is sent only if the Custom Responses option is enabled (refer to *HTTP Responses, on page 314*).
congestion_scheme	Specifies if Traffic Edge applies the rule on a per-host ("per_host") basis or on a per-IP basis ("per_ip"); for example, if the server www.host1.com has two IP addresses and you use the tag value "per_ip", each IP address has its own number of connection failures and is marked as congested independently. If you use the tag value "per_host", and the server www.host1.com is marked as congested, both IP addresses are marked as congested. The default value is "per_ip". You must enclose the value in quotes.
snmp	Configures Traffic Edge to send an SNMP trap to the configured console when a congested server is detected and again when the congestion is cleared. You can specify one of two values: "on" or "off". You must enclose the value in quotes. The default value is "on". Traffic Edge sends an SNMP trap only if the SNMP feature is enabled (refer to *Basic, on page 299*).

Examples

The following example congestion.config file rule configures Traffic Edge to stop forwarding requests to the server www.host.com on port 80 (HTTP traffic) if the server is congested according to the timeouts specified. Traffic Edge uses the default tag values because no tag has been specified.

```
dest_host=www.host.com port=80
```

You can use one or more tags in a rule but each tag must have one value only. If you specify no tags in the rule, Traffic Edge uses the default values.

You can override any of the default tag values by adding configuration variables at the end of the records.config as follows:

```
CONFIG proxy.config.http.congestion_control.default.tag  INT|STRING  value
```

tag is one of the tags described in the table on *page 371* and *value* is the value you want to use; for example:

```
CONFIG proxy.config.http.congestion_control.default.congestion_scheme  STRING
per_host
```

IMPORTANT The rules in the `congestion.config` file override the following variables in the `records.config` file:

```
proxy.config.http.connect_attempts_max_retries
proxy.config.http.connect_attempts_max_retries_dead_server
proxy.config.http.connect_attempts_rr_retries
proxy.config.http.connect_attempts_timeout
proxy.config.http.down_server.cache_time
proxy.config.http.down_server.abort_threshold
```

extensions.config

The `extensions.config` file defines the file types (based on file extensions) that you want to send to CarrierScan Server to be scanned.

IMPORTANT After you modify the `extensions.config` file, you must restart Traffic Edge.

The `extensions.config` file is located in the Traffic Edge `config/plugins` directory. The file exists only if you installed the Antivirus Extension.

Note If Traffic Edge does not contain an `extensions.config` file or if the file is empty, all file types are scanned.

Format

Each line in the `extensions.config` file must contain a file extension without the dot (.). Traffic Edge sends only files of the types listed to CarrierScan Server for scanning. The default `extensions.config` file contains a list of common file extensions. Delete the file extensions you do *not* want CarrierScan Server to scan and add those you do want CarrierScan Server to scan. To scan files with no extension, add the line `no_extension`.

IMPORTANT The list of file extensions in the `extensions.config` file must match the list of file extensions in the CarrierScan Server configuration.

Examples

The following example configures Traffic Edge to send files with the extensions `acv`, `ax`, `zip`, and `obd` and files with no extension to CarrierScan Server to be scanned. Any other file types are not scanned.

```
acv
ax
zip
obd
no_extension
```

filter.config

The `filter.config` file lets you deny or allow particular requests, strip header information from client requests, and specify LDAP, RADIUS, and NTLM authentication and group authorization rules.

IMPORTANT After you modify the `filter.config` file, you must restart Traffic Edge.

Format

Each line in the `filter.config` file contains a filtering rule. Traffic Edge applies the rules in the order listed, starting at the top of the file.

Traffic Edge recognizes three space-delimited tags:

`primary_destination=value secondary_specifier=value action=value`

The following table lists the possible primary destinations and their allowed values.

Primary Destination	Allowed Value
`dest_domain`	A requested domain name. Traffic Edge matches the domain name of the destination from the URL in the request.
`dest_host`	A requested hostname. Traffic Edge matches the hostname of the destination from the URL in the request.
`dest_ip`	A requested IP address. Traffic Edge matches the IP address of the destination in the request.
`url_regex`	A regular expression to be found in a URL.

The secondary specifiers are optional in the `filter.config` file. The following table lists the possible secondary specifiers and their allowed values.

Secondary Specifier	Allowed Value
`port`	A requested URL port.
`scheme`	A request URL protocol: http, https, ftp, rtsp, or mms.
`prefix`	A prefix in the path part of a URL.
`suffix`	A file suffix in the URL.

`method`	A request URL method. You can specify one of the following:
	`get`
	`post`
	`put`
	`trace`
	`PUSH` - If the PUSH option is enabled (the PUSH option lets you deliver content directly to the cache without user request), you must add a filtering rule with the PUSH action to ensure that only known source IP addresses implement PUSH requests to the cache. See the example below.
	You enable the PUSH option by setting the configuration variable `proxy.config.http.push_method_enabled` to 1 in the `records.config` file or by enabling the configuration option **Push Method** in Traffic Manager (**Configure/Protocols/HTTP/ General**).

Secondary Specifier	Allowed Value
`time`	A time range, such as 08:00-14:00.
`src_ip`	A client IP address. `src_ip` is not supported for PNA content. `src_ip` is not supported for RealOne Player requests when the RealProxy port is set to 554 either explicitly or transparently.
`tag`	Specifies Real Networks or QuickTime streams. You can enter `RNI` or `QT`: for example, `tag=QT`.

You can use more than one secondary specifier in a rule. However, you cannot repeat a secondary specifier.

The following table lists the possible actions and their allowed values.

Action	Allowed Value
`action=deny`	Denies requests for objects from specific destinations. When a request is denied, the client receives an access denied message.
`action=allow`	Allows particular URL requests; Traffic Edge caches and serves the requested content.
`action=radius`	Specifies RADIUS authentication rules that determine which users must be authenticated to access particular sites on the Internet and which RADIUS servers are used. You can specify the following optional parameters: ◆ `realm=`*`realm_name`* ◆ `redirect_url=`*`URL`* Traffic Edge uses the RADIUS rules only if the RADIUS option is enabled. If the RADIUS option is disabled, Traffic Edge uses LDAP authentication rules. If the LDAP option is disabled, Traffic Edge uses NTLM authentication rules. If all three proxy authentication options are disabled, Traffic Edge treats all authentication rules as `allow` rules - access is granted without authentication. For information on enabling RADIUS, refer to *Configuring Traffic Edge to Be a RADIUS Client, on page 219*.

Action	Allowed Value
`action=ntlm`	Specifies NTLM authentication and group authorization rules that determine which users must be authenticated and authorized to access particular sites on the Internet and which domain controllers are used. For NTLM authorization, you can specify the following optional parameters: ◆ `realm=`*`realm_name`* ◆ `redirect_url=`*`URL`* For NTLM group authorization, you *must* specify the following parameters with the `ntlm` action: ◆ `server=`*`server_name`*`:`*`server_port`* ◆ `dn=`*`BaseDN`* ◆ `attr=`*`additional_attribute_name`* ◆ `attr_val=`*`additional_attribute_value`* ◆ `bind_dn=`*`user_dn`* ◆ `bind_pwd_file=`*`filename`* ◆ `uid_filter=`*`UID_filter`* The `realm` and `redirect_url` parameters are optional for NTLM group authorization. Traffic Edge uses the NTLM rules only if the NTLM option is enabled. If the NTLM option is disabled, Traffic Edge uses LDAP authentication rules. If the LDAP option is disabled, Traffic Edge uses RADIUS authentication rules. If all three proxy authentication options are disabled, Traffic Edge treats all authentication rules as `allow` rules - access is granted without authentication. For information on enabling NTLM, refer to *Configuring NTLM Proxy Authentication, on page 222*.

Action	Allowed Value
`action=ldap`	Specifies LDAP authentication rules that determine which users must be authenticated to access particular sites on the Internet and which LDAP servers are used. You can specify the following optional parameters: ◆ `server=`*`LDAP_server_name`*`:`*`LDAP_server_port`* ◆ `dn=`*`BaseDN`* ◆ `uid_filter=`*`UID_filter`* ◆ `attr=`*`additional_LDAP_attribute_name`* ◆ `attr_val=`*`additional_LDAP_attribute_value`* ◆ `bind_dn=`*`user_dn`* ◆ `bind_pwd_file=`*`filename`* ◆ `realm=`*`realm_name`* ◆ *`redirect_url=URL`* The `bind_dn` and `bind_pwd_file` parameters are used for non anonymous binding to the LDAP server. If you specify the `ldap` action without any optional parameters, Traffic Edge uses the LDAP server and port listed in the `records.config` file. If you *do* use any of the `ldap` optional parameters, you must specify values for the `server`, `dn`, and `uid_filter` parameters. Traffic Edge uses the LDAP rules only if the LDAP option is enabled. If the LDAP option is disabled, Traffic Edge uses NTLM authentication rules. If the NTLM option is disabled, Traffic Edge uses RADIUS authentication rules. If all three proxy authentication options are disabled, Traffic Edge treats all authentication rules as `allow` rules; access is granted without authentication. For information on enabling LDAP, refer to *Configuring Traffic Edge to Be an LDAP Client, on page 215*

`keep_hdr`	The client request header information that you want to keep. You can specify the following options:
	`date`
	`host`
	`cookie`
	`client_ip`

`strip_hdr`	The client request header information that you want to strip. You can specify the same options as `keep_hdr`.

Examples

The following example configures Traffic Edge to deny all FTP document requests to the IP address 112.12.12.12:

```
dest_ip=112.12.12.12 scheme=ftp action=deny
```

The following example configures Traffic Edge to keep the client IP address header for URL requests that contain the regular expression `politics` and whose path prefix is `/viewpoint`:

```
url_regex=politics prefix=/viewpoint keep_hdr=client_ip
```

The following example configures Traffic Edge to strip all cookies from client requests destined for the origin server `www.server1.com`:

```
dest_host=www.server1.com strip_hdr=cookie
```

The following example configures Traffic Edge to disallow `puts` to the origin server `www.server2.com`:

```
dest_host=www.server2.com method=put action=deny
```

The following example configures Traffic Edge to allow only the host associated with the IP address 11.11.1.1 to deliver content directly into the cache (push). A deny rule is also included to prevent unauthorized users from pushing content into the cache.

```
dest_domain=. src_ip=11.11.1.1 method=PUSH action=allow
dest_domain=. method=PUSH action=deny
```

The following example configures Traffic Edge to allow access to the origin server `www.server.com` only to users authenticated by the LDAP server running on `ldap.com`:

```
dest_host=www.server.com action=ldap server=ldap.com dn="o=ldap.com"
```

The following example requires all users to be authenticated by the default LDAP server (the LDAP server listed in the `records.config` file):

```
dest_ip=0.0.0.0-255.255.255.255 action=ldap
```

Traffic Edge applies the rules in the order listed in the `filter.config` file; for example, the sample `filter.config` file below configures Traffic Edge to do the following:

- Allow only users authenticated by the LDAP server on `ldap.com` to access `internal.com`
- Allow all users (except those trying to access `internal.com`) to access `server1.com`
- Deny all users access to `playboy.com`
- Allow authenticated users (authenticated by the LDAP server on `ldap.com`) with the attribute ou="Accounting Department" in their LDAP profile to access `401kfidelity.com`
- Require any user not included in any of the above directives to be authenticated by the default LDAP server

```
dest_host=internal.com  action=ldap  server=ldap.com  dn="o=ldap.com"
dest_host=server1.com  action=allow
dest_host=playboy.com  action=deny
dest_host=401k.fidelity.com  action=ldap  server=ldap.com:389
dn="o=ldap.com"  attr=ou  attr_val="Accounting Department"
dest_ip=0.0.0.0-255.255.255.255  action=ldap
```

ftp_remap.config

The `ftp_remap.config` file is used for FTP reverse proxy. This file contains mapping rules that Traffic Edge uses to direct any incoming FTP requests to the FTP server if the requested document is a cache miss or is stale.

IMPORTANT After you modify the `ftp_remap.config` file, navigate to the Traffic Edge `bin` directory (in Windows, open a Command Prompt window and navigate to the `Traffic Edge bin` directory) and then run the `traffic_line -x` command to apply the changes. When you apply the changes to a node in a cluster, Traffic Edge automatically applies the changes to all other nodes in the cluster.

For information about using and configuring FTP reverse proxy, refer to *Chapter 7, Reverse Proxy and HTTP Redirects*.

Format

Each line in the `ftp_remap.config` file contains a mapping rule in the format:

`Traffic_Edge:port ftp_server:port`

`Traffic_Edge` is the IP address or hostname assigned to Traffic Edge, *`ftp_server`* is the IP address or hostname assigned to the FTP server to which you want to redirect requests, and *`port`* is the port number.

Examples

The following example configures Traffic Edge to direct all FTP requests sent to the Traffic Edge IP address 111.111.11.1 to the FTP server IP address 11.11.11.1:

```
111.111.11.1:7999 11.11.11.1:21
```

hosting.config

The `hosting.config` file lets you assign cache partitions to specific origin servers and/or domains so that you can manage your cache space more efficiently and restrict disk usage.

For step by step instructions on partitioning the cache according to origin servers and/or domains, refer to *Partitioning the Cache According to Origin Server or Domain, on page 168*.

Before you can assign cache partitions to specific origin servers and/or domains, you must partition your cache according to size and protocol in the `partition.config` file. For step-by-step instructions on partitioning your cache, refer to *Partitioning the Cache, on page 167*. For a description of the `partition.config` file, refer to *partition.config, on page 396*.

After you modify the `hosting.config` file, navigate to the Traffic Edge `bin` directory (in Windows, open a Command Prompt window and navigate to the `Traffic Edge bin` directory) and then run the `traffic_line -x` command to apply the changes. When you apply the changes to a node in a cluster, Traffic Edge automatically applies the changes to all other nodes in the cluster.

IMPORTANT The partition configuration must be the same on all nodes in a cluster.

Format

Each line in the `hosting.config` file must have one of the following formats:

`hostname=`*`hostname`* `partition=`*`partition_numbers`*

`domain=`*`domain_name`* `partition=`*`partition_numbers`*

`hostname` is the fully qualified hostname of the origin server whose content you want to store on a particular partition: for example, `www.myhost.com`.

`domain_name` is the domain whose content you want to store on a particular partition: for example, `mydomain.com`.

`partition_numbers` is a comma-separated list of the partitions on which you want to store the content that belongs to the origin server or domain listed. The partition numbers must be valid numbers listed in the `partition.config` file (refer to *partition.config, on page 396*).

Note If you want to allocate more than one partition to an origin server or domain, you must enter the partitions in a comma-separated list on one line, as shown in the example below. The `hosting.config` file cannot contain multiple entries for the same origin server or domain.

Generic Partition

When configuring the `hosting.config` file, you must assign a generic partition to use for content that does not belong to any of the origin servers or domains listed. If all partitions for a particular origin server become corrupt, Traffic Edge will also use the generic partition to store content for that origin server.

The generic partition must have the following format:

`hostname=* partition=`*`partition_numbers`*

`partition_numbers` is a comma-separated list of generic partitions.

Examples

The following example configures Traffic Edge to store content from the domain `mydomain.com` in partition 1 and content from `www.myhost.com` in partition 2. Traffic Edge stores content from all other origin servers in partitions 3 and 4.

```
domain=mydomain.com partition=1
hostname=www.myhost.com partition=2
hostname=* partition=3,4
```

icp.config

The `icp.config` file defines ICP peers (parent and sibling caches).

IMPORTANT After you modify the `icp.config` file, navigate to the Traffic Edge `bin` directory (in Windows, open a Command Prompt window and navigate to the `Traffic Edge bin` directory) and then run the `traffic_line -x` command to apply the changes. When you apply the changes to a node in a cluster, Traffic Edge automatically applies the changes to all other nodes in the cluster.

Format

Each line in the `icp.config` file contains the name and configuration information for a single ICP peer in the following format:

host:*host_IP*:*peer_type*:*proxy_port*:*icp_port*:*MC_on*:*MC_IP*:*MC_TTL*:

Each field is described in the following table.

Field	Description
host	The hostname of the ICP peer. This field is optional; if you do not specify the hostname of the ICP peer, you must specify the IP address.
host _IP	The IP address of the ICP peer. This field is optional; if you do not specify the IP address of the ICP peer, you must specify the hostname.
ctype	Use the following options: `1` to indicate an ICP parent cache `2` to indicate an ICP sibling cache
proxy_port	The port number of the TCP port used by the ICP peer for proxy communication.
icp_port	The port number of the UDP port used by the ICP peer for ICP communication.
multicast_member	Multicast on/off. Use the following options: `0` if multicast is disabled `1` if multicast is enabled

Field	Description
multicast_ip_str	The multicast IP address.
multicast_ttl	The multicast time to live. Use the following options: 1 if IP multicast datagrams will not be forwarded beyond a single subnetwork 2 to allow delivery of IP multicast datagrams to more than one subnet (if there are one or more multicast routers attached to the first hop subnet)

Examples

The following example configuration is for three nodes: the local host, one parent, and one sibling.

```
localhost:0.0.0.0:3:8080:3130:0:0.0.0.0:1
host1:123.12.1.23:1:8080:3131:0:0.0.0.0:1
host2:123.12.1.24:2:8080:3131:0:0.0.0.0:1
```

ip_allow.config

The `ip_allow.config` file controls client access to the Traffic Edge proxy cache. You can specify ranges of IP addresses that are allowed to use the Traffic Edge as a web proxy cache.

IMPORTANT After you modify the `ip_allow.config` file, navigate to the Traffic Edge `bin` directory (in Windows, open a Command Prompt window and navigate to the `Traffic Edge bin` directory) and then run the `traffic_line -x` command to apply the changes. When you apply the changes to a node in a cluster, Traffic Edge automatically applies the changes to all other nodes in the cluster.

Format

Each line in the `ip_allow.config` file must have the following format:

```
src_ip=ipaddress action=ip_allow | ip_deny
```

ipaddress is the IP address or range of IP addresses of the clients allowed to access the Traffic Edge proxy cache.

The action `ip_allow` allows the specified clients to access the Traffic Edge proxy cache.

The action `ip_deny` denies the specified clients to access the Traffic Edge proxy cache.

By default, the `ip_allow.config` file contains the following line, which allows all clients to access the Traffic Edge proxy cache. Comment out or delete this line before adding rules to restrict access.

```
src_ip=0.0.0.0-255.255.255.255 action=ip_allow
```

Examples

The following example allows all clients to access the Traffic Edge proxy cache:

```
src_ip=0.0.0.0-255.255.255.255 action=ip_allow
```

The following example allows all clients on a specific subnet to access the Traffic Edge proxy cache:

```
src_ip=123.12.3.000-123.12.3.123 action=ip_allow
```

The following example denies all clients on a specific subnet to access the Traffic Edge proxy cache:

```
src_ip=123.45.6.0-123.45.6.123 action=ip_deny
```

ipnat.conf

The `ipnat.conf` file contains redirection rules that specify how incoming packets are readdressed when Traffic Edge is serving traffic transparently. Traffic Edge creates the redirection rules during installation. You can modify these rules to suit your needs.

IMPORTANT After you modify the `ipnat.conf` file, you must restart Traffic Edge.

Format

Each line in the `ipnat.config` file must have the following format:

`rdr` *`interface`* `0.0.0.0/0 port` *`dest`* `->` *`ipaddress`* `port` *`proxy`* `tcp|udp`

`interface` is the Ethernet interface that traffic will use to access the Traffic Edge machine: for example, `hme0` on Solaris or `eth0` on Linux.

`dest` is the traffic destination port: for example, 80 for HTTP traffic.

`ipaddress` is the IP address of your Traffic Edge.

`proxy` is the Traffic Edge proxy port, usually 8080 for HTTP traffic.

Examples

The following example configures the Traffic Edge ARM to readdress all incoming HTTP traffic to the Traffic Edge IP address (111.111.11.1) on proxy port 8080:

```
rdr hme0  0.0.0.0/0  port 80 -> 111.111.11.1  port  8080  tcp
```

logs.config

The `logs.config` file establishes and formats *traditional* custom transaction log files.

IMPORTANT Previous Traffic Edge releases provide traditional custom logging in addition to the XML-based custom logging. Although this release of Traffic Edge continues to support traditional custom logging, Inktomi recommends that you use the XML-based custom logging, which is more versatile. Refer to *Using the Custom Format, on page 246*, and *logs_xml.config, on page 387*.

If you opt to use traditional custom logging instead of the more versatile XML-based custom logging, you must enable the traditional custom logging option manually. Refer to *Support for Traditional Custom Logging, on page 277*.

IMPORTANT After you modify the `logs.config` file, navigate to the Traffic Edge `bin` directory (in Windows, open a Command Prompt window and navigate to the `Traffic Edge bin` directory) and then run the `traffic_line -x` command to apply the changes. When you apply the changes to a node in a cluster, Traffic Edge automatically applies the changes to all other nodes in the cluster.

Format

Each line in the `logs.config` file establishes and formats a custom transaction log file. Lines consist of the following fields, separated by colons (:).

Field	Allowed Inputs
`format`	All lines must begin with the word `format`.

`activation flag`	Specifies if the custom log file is activated. You can specify one of the following options:
	`enabled`
	`disabled`

`unique format identifier`	You must use a unique integer for each custom log file you create.
`format name`	Specifies the name for the format you define.
`format string`	Identifies the `printf`-style format string specifying the field symbols to be displayed and how they should look in ASCII. Refer to *Appendix D* for a list of the available field symbols and their meanings. Field symbols are indicated by %<*`field_symbol`*> format; for example, to indicate that `chi` is the client host IP and not the string `chi` to be printed, enter %<chi>.
`file name`	Specifies the name of the custom log file you create.
`file type`	Specifies the format of the file:`ASCII` or `BINARY`.
`file header data`	Specifies the header text. Enter `none` if you do not want header text. Enter the text of the header if you want your custom log file to have a header.

Examples

The following example shows a custom log format for a file named `minimalist`. It records the client host IP address (`chi`), the client request universal resource identifier (`cqu`), and the proxy response status code (`pssc`)

```
format:enabled:1:minimal:%<chi> / %<cqu> / %<pssc>:minimalist:ASCII:none
```

The output file for the above example format is:

```
123.12.3.123 / GET http://earth/ocean/index.html HTTP/1.0 / 200
```

The following example shows a custom log format for a file named `test`. It records the User-Agent value of the client request header (`cqh`) and the Retry-After value of the proxy response header (`psh`).

```
format:enabled:1:test:%<{User-Agent}cqh> %<{Retry-After}psh>:test:ASCII:none
```

WELF

Traffic Edge supports WELF, the WebTrends Enhanced Log format, so that you can analyze Traffic Edge log files with WebTrends reporting tools. A predefined custom format for WELF is provided in the `logs.config` file. To create a WELF format log file, comment out the following section at the end of the file and replace `<FORMAT_ID>` with a unique integer.

```
#format:enabled:<FORMAT_ID>:welf:id=firewall time="%<cqtd> %<cqtt>"
fw=%<phn> pri=6 proto=%<cqus> duration=%<ttmsf> sent=%<psql> rcvd=%<cqhl>
src=%<chi> dst=%<shi> dstname=%<shn> user=%<caun> op=%<cqhm>
arg="%<cqup>" result=%<pssc> ref="%<{Referer}cqh>" agent="%<{user-
agent}cqh>" cache=%<crc>:welf:ASCII:none
#
```

log_hosts.config

To record HTTP/FTP transactions for different origin servers in separate log files, you must list each origin server hostname in the `log_hosts.config` file. In addition, you must enable the HTTP host splitting option (refer to *HTTP Host Log Splitting, on page 255*).

Inktomi recommends that you use the same `log_hosts.config` file on every Traffic Edge node in your cluster.

IMPORTANT After you modify the `log_hosts.config` file, navigate to the Traffic Edge `bin` directory (in Windows, open a Command Prompt window and navigate to the `Traffic Edge bin` directory) and then run the `traffic_line -x` command to apply the changes. When you apply the changes to a node in a cluster, Traffic Edge automatically applies the changes to all other nodes in the cluster.

Format

Each line in the `log_hosts.config` file has the following format:

`hostname`

`hostname` is the hostname of the origin server.

Tip You can specify keywords in the `log_hosts.config` file to record all transactions from origin servers with the specified keyword in their names in a separate log file. See the example below.

Examples

The following example configures Traffic Edge to create separate log files containing all HTTP/FTP transactions for the origin servers `webserver1`, `webserver2`, and `webserver3`.

```
webserver1
webserver2
webserver3
```

The following example records all HTTP and FTP transactions from origin servers that contain `sports` in their names: for example, `sports.yahoo.com` and `www.foxsports.com`, in a log file called `squid-sport.log` (the Squid format is enabled):

```
sports
```

logs_xml.config

The `logs_xml.config` file defines the custom log file formats, filters, and processing options. The format of this file is modeled after XML, the Extensible Markup Language.

Format

The `logs_xml.config` file contains the following specifications:

- `LogFormat` specifies the fields to be gathered from each protocol event access.
- `LogFilter` specifies the filters that are used to include or exclude certain entries being logged based on the value of a field within that entry.
- `LogObject` specifies an object that contains a particular format, a local filename, filters, and collation servers.

The `logs_xml.config` file ignores extra white space, blank lines, and all comments.

LogFormat

The following table lists the `LogFormat` specifications.

Field	Allowed Inputs
<Name = "valid_format_name"/>	Required. Valid format names include any name except squid, common, extended, or extended2, which are pre-defined formats. There is no default for this tag.
<Format = "valid_format_specification"/>	Required. A valid format specification is a printf-style string describing each log entry when formatted for ASCII output. Use '%<*field*>' as placeholders for valid Inktomi field names. For more information, refer to *Inktomi Custom Logging Fields, on page 359*. The specified field can be one of the following types: Simple: for example, %<cqu> A field within a container, such as an HTTP header or an Inktomi statistic. Fields of this type have the syntax: '%<{*field*}*container*>'. Aggregates, such as COUNT, SUM, AVG, FIRST, LAST. Fields of this type have the syntax: '%<operator (*field*)>'. Note: You cannot create a format specification that contains both aggregate operators and regular fields.

<Interval = "aggregate_interval_secs"/>	Use this tag when the format contains aggregate operators. The value "aggregate_interval_secs" represents the number of seconds between individual aggregate values being produced. The valid set of aggregate operators are:
	`COUNT`
	`SUM`
	`AVG`
	`FIRST`
	`LAST`

LogFilters

The following table lists the `LogFilter` specifications.

Field	Allowed Inputs
<Name = "valid_filter_name"/>	Required. All filters must be uniquely named.
<Condition = “valid_log_field valid_operator valid_comparison_value”/>	Required. This field contains the following elements: *`valid_log_field`* - the field that will be compared against the given value. For more information, refer to *Logging Format Cross-Reference, on page 363*. *`valid_operator_field`* - any one of the following: `MATCH`, `CASE_INSENSITIVE_MATCH`, `CONTAIN`, `CASE_INSENSITIVE_CONTAIN`. `MATCH` is true if the field and value are identical (case sensitive). `CASE_INSENSITIVE_MATCH` is similar to `MATCH`, only case insensitive. `CONTAIN` is true if the field contains the value (the value is a substring of the field). `CASE_INSENSITIVE_CONTAIN` is a case-insensitive version of `CONTAIN`. *`valid_comparison_value`* - any string or integer matching the field type. For integer values, all of the operators are equivalent and mean that the field must be equal to the specified value. Note: There are no negative comparison operators. If you want to specify a negative condition, use the `Action` field to `REJECT` the record.
<Action = “valid_action_field”/>	Required. `ACCEPT` or `REJECT`. This instructs Traffic Edge to either accept or reject records satisfying the condition of the filter.

LogObject

The following table lists the `LogObject` specifications.

Field	Allowed Inputs
<Format = “valid_format_name”/>	Required. Valid format names include the predefined logging formats: squid, common, extended, and extended2, as well as any previously-defined custom log formats. There is no default for this tag.
<Filename = “file_name”/>	Required. The filename to which the given log file is written on the local file system or on a remote collation server. No local log file will be created if you fail to specify this tag. All filenames are relative to the default logging directory. If the name does not contain an extension: for example, `squid`, the extension `.log` is automatically appended to it for ASCII logs and `.blog` for binary logs. (Refer to *<Mode = “valid_logging_mode”/>*.) If you do not want an extension to be added, end the filename with a single dot (.): for example, `squid.`
<Mode = “valid_logging_mode”/>	Valid logging modes include `ascii`, `binary`, and `ascii_pipe`. The default is `ascii`. Use `ascii` to create event log files in human-readable form (plain ASCII). Use `binary` to create event log files in binary format. Binary log files generate lower system overhead and occupy less space on the disk (depending on the information being logged). You must use the logcat utility to translate binary log files to ASCII format before you can read them. Use `ascii_pipe` to write log entries to a UNIX named pipe (a buffer in memory). Other processes can then read the data using standard I/O functions. The advantage of using this option is that Traffic Edge does not have to write to disk, freeing disk space and bandwidth for other tasks. In addition, writing to a pipe does not stop when logging space is exhausted because the pipe does not use disk space. If you are using a collation server, the log is written to a pipe on the collation server. A local pipe is created even before a transaction is processed so that you can see the pipe right after Traffic Edge starts. However, pipes on a collation server *are* created when Traffic Edge starts.
<Filters = “list_of_valid_filter_names”/>	A comma-separated list of names of any previously defined log filters. If more than one filter is specified, all filters must accept a record for the record to be logged.
<Protocols = “list_of_valid_protocols”/>	A comma-separated list of the protocols this object should log. Valid protocol names include HTTP, FTP, and ICP.

Field	Allowed Inputs
<ServerHosts = “list_of_valid_servers”/>	A comma-separated list of valid hostnames.This tag indicates that only entries from the named servers will be included in the file.
<CollationHosts = “list_of_valid_hostnames”/>	A comma-separated list of collation servers to which all log entries (for this object) are forwarded. Collation servers can be specified by name or IP address. Specify the collation port with a colon after the name: for example, `host:port`.
<Header = "header"/>	The header text you want the log files to contain. The header text appears at the beginning of the log file, just before the first record.
<RollingEnabled = "truth value"/>	Enables or disables log file rolling for the LogObject. This setting overrides the value for the configuration setting **Log Rolling: Enabled/Disabled** in Traffic Manager or `proxy.config.log2.rolling_enabled` in the `records.config` file. Set `"truth value"` to one of the following values: `0` to disable rolling for this particular LogObject. `1` to roll log files at specific intervals during the day (you specify time intervals with the `RollingIntervalSec` and `RollingOffsetHr` fields). 2 to roll log files when they reach a certain size (you specify the size with the`RollingSizeMb` field). 3 to roll log files at specific intervals during the day or when they reach a certain size (whichever occurs first). 4 to roll log files at specific intervals during the day when log files reach a specific size (at a specified time if the file is of the specified size).
<RollingIntervalSec = "seconds"/>	Specifies the seconds between log file rolling for the LogObject. This setting overrides the value for the configuration setting **Log Rolling: Interval** in Traffic Manager or `proxy.config.log2.rolling_interval_sec` in the `records.config` file. This option allows you to specify different rolling intervals for different LogObjects.
<RollingOffsetHr = "hour"/>	Specifies an hour (from 0 to 23) at which rolling is guaranteed to align. Rolling might start before then, but a rolled file will be produced only at that time. The impact of this setting is only noticeable if the rolling interval is larger than one hour. This setting overrides the configuration setting **Log Rolling: Offset Hour** in Traffic Manager or `proxy.config.log2.rolling_offset_hr` in the `records.config` file.
<RollingSizeMb = "size_in_MB"/>	Specifies the size at which log files are rolled.

Examples

The following is an example of a `LogFormat` specification collecting information using three common fields:

```
<LogFormat>
  <Name = "minimal"/>
  <Format = "%<chi> : %<cqu> : %<pssc>"/>
</LogFormat>
```

The following is an example of a `LogFormat` specification using aggregate operators:

```
<LogFormat>
  <Name = "summary"/>
  <Format = "%<LAST(cqts)> : %<COUNT(*)> : %<SUM(psql)>"/>
  <Interval = "10"/>
</LogFormat>
```

The following is an example of a `LogFilter` that will cause only `REFRESH_HIT` entries to be logged:

```
<LogFilter>
  <Name = "only_refresh_hits"/>
  <Action = "ACCEPT"/>
  <Condition = "%<pssc> MATCH REFRESH_HIT"/>
</LogFilter>
```

Note When specifying the field in the filter condition, you can omit the `%<>`. This means that the following filter is equivalent to the example directly above:

```
<LogFilter>
  <Name = "only_refresh_hits"/>
  <Action = "ACCEPT"/>
  <Condition = "pssc MATCH REFRESH_HIT"/>
</LogFilter>
```

The following is an example of a `LogObject` specification that creates a local log file for the minimal format defined earlier. The log filename will be `minimal.log` because this is an ASCII log file (the default).

```
<LogObject>
  <Format = "minimal"/>
  <Filename = "minimal"/>
</LogObject>
```

The following is an example of a `LogObject` specification that includes only HTTP requests served by hosts in the domain `company.com` or by the specific server `server.somewhere.com`. Log entries are sent to port 4000 of the collation host `logs.company.com` and to port 5000 of the collation host 209.131.52.129.

```
<LogObject>
  <Format = "minimal"/>
  <Filename = "minimal"/>
  <ServerHosts = "company.com,server.somewhere.com"/>
```

```
  <Protocols = "http"/>
  <CollationHosts = "logs.company.com:4000,209.131.52.129:5000"/>
</LogObject>
```

WELF

Traffic Edge supports WELF, the WebTrends Enhanced Log format, so that you can analyze Traffic Edge log files with WebTrends reporting tools. A predefined `<LogFormat>` that is compatible with WELF is provided at the end of the `logs.config` file (shown below). To create a WELF format log file, create a `<LogObject>` that uses this predefined format.

```
<LogFormat>
  <Name = "welf"/>
  <Format = "id=firewall time=\"%<cqtd> %<cqtt>\" fw=%<phn> pri=6
proto=%<cqus> duration=%<ttmsf> sent=%<psql> rcvd=%<cqhl> src=%<chi>
dst=%<shi> dstname=%<shn> user=%<caun> op=%<cqhm> arg=\"%<cqup>\"
result=%<pssc> ref=\"%<{Referer}cqh>\" agent=\"%<{user-agent}cqh>\"
cache=%<crc>"/>
</LogFormat>
```

mgmt_allow.config

The `mgmt_allow.config` file specifies the IP addresses of remote hosts allowed access or denied access to Traffic Manager.

IMPORTANT After you modify the `mgmt_allow.config` file, navigate to the Traffic Edge `bin` directory (in Windows, open a Command Prompt window and navigate to the `Traffic Edge bin` directory) and then run the `traffic_line -x` command to apply the changes. When you apply the changes to a node in a cluster, Traffic Edge automatically applies the changes to all other nodes in the cluster.

Format

Each line in the `mgmt_allow.config` file has the following format:

`src_ip=`*`ipaddress`* `action=ip_allow|ip_deny`

`ipaddress` is the IP address or range of IP addresses allowed to access Traffic Manager.

`action` must specify either `ip_allow` to allow access to Traffic Manager or `ip_deny` to deny access to Traffic Manager.

By default, the `mgmt_allow.config` file contains the following line, which allows all remote hosts to access Traffic Manager. Comment out or delete this line before adding rules to restrict access.

```
src_ip=0.0.0.0-255.255.255.255 action=ip_allow
```

Examples

The following example configures Traffic Edge to allow only one user to access Traffic Manager:

```
src_ip=123.12.3.123 action=ip_allow
```

The following example configures Traffic Edge to allow a range of IP addresses to access Traffic Manager:

```
src_ip=123.12.3.000-123.12.3.123 action=ip_allow
```

The following example configures Traffic Edge to deny the IP address `123.45.67.8` access to Traffic Manager:

```
src_ip=123.45.67.8 action=ip_deny
```

parent.config

The `parent.config` file identifies the parent proxies used in an cache hierarchy. Use this file to perform the following configuration:

- Set up parent cache hierarchies, with multiple parents and parent failover
- Configure selected URL requests to bypass parent proxies

Traffic Edge uses the `parent.config` file only when the parent caching option is enabled; refer to *Configuring Traffic Edge to Use a Parent Cache, on page 161*.

IMPORTANT After you modify the `parent.config` file, navigate to the Traffic Edge `bin` directory (in Windows, open a Command Prompt window and navigate to the `Traffic Edge bin` directory) and then run the `traffic_line -x` command to apply the changes. When you apply the changes to a node in a cluster, Traffic Edge automatically applies the changes to all other nodes in the cluster.

Format

Each line in the `parent.config` file must contain a parent caching rule. Traffic Edge recognizes three space-delimited tags:

`primary_destination=value secondary_specifier=value action=value`

The following table lists the possible primary destinations and their allowed values.

Primary Destination	Allowed Value
`dest_domain`	A requested domain name.
`dest_host`	A requested hostname.
`dest_ip`	A requested IP address or range of IP addresses separated by a dash (-).
`url_regex`	A regular expression to be found in a URL

The secondary specifiers are optional in the `parent.config` file. The following table lists the possible secondary specifiers and their allowed values.

Secondary Specifiers	Allowed Value
`port`	A requested URL port.
`scheme`	A request URL protocol: http, https, ftp, rtsp, or mms.
`prefix`	A prefix in the path part of a URL.
`suffix`	A file suffix in the URL.

`method`	A request URL method; one of the following:
	`get`
	`post`
	`put`
	`trace`

`time`	A time range, such as 08:00-14:00, during which the parent cache is used to serve requests.
`src_ip`	A client IP address.
`tag`	Specifies Real Networks or QuickTime streams. You can enter `RNI` or `QT`: for example, `tag=QT`.

The following tables lists the possible actions and their allowed values.

Action	Allowed Value
`parent`	An ordered list of parent servers. If the request cannot be handled by the last parent server in the list, it will be routed to the origin server. You can specify either a hostname or an IP address. You must specify the port number.
`round_robin`	One of the following values: `true` - Traffic Edge goes through the parent cache list in a round-robin based on client IP address. `strict` - Traffic Edge machines serve requests strictly in turn; for example, machine `proxy1` serves the first request, `proxy2` serves the second request, and so on. `false` - round-robin selection does not occur.
`go_direct`	One of the following values: `true` - requests bypass parent hierarchies and go directly to the origin server. `false` - requests do not bypass parent hierarchies.

Examples

The following rule configures a parent cache hierarchy consisting of Traffic Edge (which is the child) and two parents, `p1.x.com` and `p2.x.com`. Traffic Edge forwards the requests it cannot serve to the parent servers `p1.x.com` and `p2.x.com` in a round-robin fashion because `round_robin=true`,.

```
dest_domain=. method=get parent="p1.x.com:8080; p2.y.com:8080" round_robin=true
```

The following rule configures Traffic Edge to route all requests containing the regular expression `politics` and the path `/viewpoint` directly to the origin server (bypassing any parent hierarchies):

```
url_regex=politics prefix=/viewpoint go_direct=true
```

Every line in the `parent.config` file must contain either a `parent=` or `go_direct=` directive.

partition.config

The `partition.config` file lets you manage your cache space more efficiently and restrict disk usage by creating cache partitions of different sizes for specific protocols. You can further configure these partitions to store data from certain origin servers and/or domains in the `hosting.config` file (refer to *hosting.config, on page 381*).

For step-by-step instructions on partitioning the cache, refer to *Partitioning the Cache, on page 167*.

IMPORTANT The partition configuration must be the same on all nodes in a cluster.

You must stop Traffic Edge before you change the cache partition size and protocol assignment.

Format

For each partition you want to create, enter a line with the following format:

`partition=`*`partition_number`* `scheme=`*`protocol_type`* `size=`*`partition_size`*

`partition_number` is a number between 1 and 255 (the maximum number of partitions is 255).

`protocol_type` is either `http` or `mixt` (all streaming media content is stored in the `mixt` partition and all other content is stored in the `http` partition).

Traffic Edge supports two different partition types: *http* for HTTP and FTP content and *mixt* for QuickTime and WMT streaming media content. Traffic Edge provides these two partition types to support the different size and service requirements for streaming media objects.

IMPORTANT Traffic Edge does not use the cache to store streams for Real Networks but uses the filesystem instead; refer to the *Traffic Edge Installation Guide* for more information.

`partition_size` is the amount of cache space allocated to the partition. This value can be either a percentage of the total cache space or an absolute value. The absolute value must be a multiple of 128 MB, where 128 MB is the smallest value. If you specify a percentage, the size is rounded down to the closest multiple of 128 MB.

Each partition is striped across several disks to achieve parallel I/O; for example, if there are four disks, a one gigabyte partition will have 256 MB on each disk (assuming each disk has enough free space available).

Note If you do not allocate all the disk space in the cache, the extra disk space is not used. You can use the extra space later to create new partitions without deleting and clearing the existing partitions.

Examples

The following example partitions the cache evenly between the HTTP and the QuickTime and WMT streaming media requests:

```
partition=1 scheme=http size=50%
partition=2 scheme=mixt size=50%
```

records.config

The `records.config` file is a list of configurable variables that Traffic Edge software uses. This section describes these variables.

Many of the variables in the `records.config` file are set automatically when you set configuration options in Traffic Manager, Traffic Line, or Traffic Shell. Certain configuration options can be set only by editing variables manually in the `records.config` file.

Warning Do not change the `records.config` variables unless you are certain of the effect. Many variables are coupled, meaning that they interact with other variables. Changing a single variable in isolation could cause the Traffic Edge to fail. Whenever possible, use Traffic Manager or Traffic Line to configure Traffic Edge.

IMPORTANT After you modify the `records.config` file, navigate to the Traffic Edge `bin` directory (in Windows, open a Command Prompt window and navigate to the `Traffic Edge bin` directory) and then run the `traffic_line -x` command to apply the changes. When you apply the changes to a node in a cluster, Traffic Edge automatically applies the changes to all other nodes in the cluster.

Format

Each variable has the following format:

```
CONFIG variable_name DATATYPE variable_value
```

DATATYPE is `INT` (an integer), `STRING` (a string), or `FLOAT` (a floating point).

Examples

In the following example, the variable `proxy.config.proxy_name` is of datatype string and its value is `my_server`. This means that the name of the Traffic Edge proxy is `my_server`.

```
CONFIG proxy.config.proxy_name STRING my_server
```

In the following example, the variable `proxy.config.arm.enabled` is a yes/no flag. A value of `0` (zero) disables the option. A value of 1 enables the option.

```
CONFIG proxy.config.arm.enabled INT 0
```

In the following example, the variable sets the cluster startup timeout to 10 seconds.

```
CONFIG proxy.config.cluster.startup_timeout INT 10
```

Configuration Variables

The following table describes the configuration variables listed in the `records.config` file.

Configuration Variable Data Type	Default Value	Description
System Variables		
proxy.config.proxy_name STRING		Specifies the name of the Traffic Edge node.
proxy.config.bin_path STRING	bin	Specifies the location of the Traffic Edge `bin` directory.
proxy.config.proxy_binary STRING	traffic_server (UNIX) traffic_server.exe (Windows)	Specifies the name of the executable that runs the traffic_server process.
proxy.config.proxy_binary_opts STRING	-M	Specifies the command-line options for starting Traffic Edge.
proxy.config.manager_binary STRING	traffic_manager (UNIX) traffic_manager.exe (Windows)	Specifies the name of the executable that runs the traffic_manager process.
proxy.config.cli_binary STRING	traffic_line (UNIX) traffic_line.exe (Windows)	Specifies the name of the executable that runs the command-line interface (Traffic Line).
proxy.config.watch_script STRING	traffic_cop	Specifies the name of the executable that runs the traffic_cop process.
proxy.config.env_prep STRING	example_prep.sh (UNIX) example_prep.bat (Windows)	Specifies the script that is executed before the traffic_manager process spawns the traffic_server process.
proxy.config.config_dir STRING	config	Specifies the directory that contains the Traffic Edge configuration files.
proxy.config.temp_dir STRING	/tmp	Specifies the directory used for Traffic Edge temporary files (UNIX only.) In Windows, the TEMP environment variable is used instead.

Configuration Variable Data Type	Default Value	Description
proxy.config.alarm_email STRING	inktomi	Specifies the email address to which Traffic Edge sends alarm messages. During a custom Traffic Edge installation, you can specify the email address; otherwise, Traffic Edge uses the Traffic Edge user account name as the default value for this variable.
proxy.config.syslog_facility STRING	LOG_DAEMON	Specifies the facility used to record system log files (UNIX only.) Refer to *Understanding Traffic Edge Log Files, on page 240*.
proxy.config.cop.core_signal INT	0	Specifies the signal that is sent to traffic_cop's managed processes to stop them. UNIX only. 0 = no signal is sent.
proxy.config.cop.linux_min_swapfree_kb INT	10240	Specifies the minimum amount of free swap space allowed before Traffic Edge stops the traffic_server and traffic_manager processes to prevent the system from hanging. This configuration variable applies if swap is enabled in Linux 2.2 only.
proxy.config.cop.linux_min_memfree_kb INT	10240	Specifies the minimum amount of free memory allowed before Traffic Edge stops the traffic_server and traffic_manager processes to prevent the system from hanging. This configuration variable applies if swap is *disabled* in Linux 2.2 only.
proxy.config.output.logfile STRING	traffic.out	Specifies the name and location of the file that contains warnings, status messages, and error messages produced by the Traffic Edge processes. If no path is specified, Traffic Edge creates the file in its logging directory.

Configuration Variable Data Type	Default Value	Description
proxy.config.snapshot_dir STRING	snapshots	Specifies the directory in which Traffic Edge stores configuration snapshots on the local system. Unless you specify an absolute path, this directory is located in the Traffic Edge `config` directory.
Local Manager		
proxy.config.lm.sem_id INT	11452	Specifies the semaphore ID for the local manager.
proxy.local.cluster.type INT	3	Sets the clustering mode: 1 = full-clustering mode 2 = management-only mode 3 = no clustering
proxy.config.cluster.rsport INT	8088	Specifies the reliable service port. The reliable service port is used to send configuration information between the nodes in a cluster. All nodes in a cluster must use the same reliable service port.
proxy.config.cluster.mcport INT	8089	Specifies the multicast port. The multicast port is used for node identification. All nodes in a cluster must use the same multicast port.
proxy.config.cluster.mc_group_addr STRING	224.0.1.37	Specifies the multicast address for cluster communications. All nodes in a cluster must use the same multicast address.
proxy.config.cluster.mc_ttl INT	1	Specifies the multicast Time to Live for cluster communications.
proxy.config.cluster.log_bogus_mc_msgs INT	1	Enables (1) or disables (0) logging of bogus multicast messages.
proxy.config.admin.html_doc_root STRING	ui	Specifies the document root for Traffic Manager.
proxy.config.admin.web_interface_port INT	8081	Specifies the Traffic Manager port.
proxy.config.admin.autoconf_port INT	8083	Specifies the autoconfiguration port.
proxy.config.admin.overseer_port INT	8082	Specifies the port used for retrieving and setting statistics and configuration variables.

Configuration Variable Data Type	Default Value	Description
proxy.config.admin.admin_user STRING	admin	Specifies the administrator ID that controls access to Traffic Manager.
proxy.config.admin.admin_password STRING		Specifies the encrypted administrator password that controls access to Traffic Manager. You cannot edit the password; however, you can specify a value of `NULL` to clear the password. Refer to *How do you access Traffic Manager if you forget the master administrator password?, on page 475*.
proxy.config.admin.basic_auth INT	1	Enables (1) or disables (0) basic user authentication to control access to Traffic Manager. Note: If basic authentication is disabled, any user can access the Traffic Manager to monitor and configure Traffic Edge.
proxy.config.admin.use_ssl INT	0	Enables the Traffic Manager SSL option for secure communication between a remote host and the Traffic Manager.
proxy.config.admin.ssl_cert_file STRING	private_key.pem	Specifies the filename of the SSL certificate installed on the Traffic Edge system for secure communication between a remote host and Traffic Manager.
proxy.config.admin.number_config_bak INT	3	Specifies the maximum number of copies of rolled configuration files to keep.
proxy.config.admin.user_id STRING	inktomi	Specifies the nonprivileged user account designated to Traffic Edge (UNIX only.)
proxy.config.admin.ui_refresh_rate INT	30	Specifies the refresh rate for the display of statistics in the Traffic Manager Monitor pages.
proxy.config.admin.log_mgmt_access INT	0	Enables (1) or disables (0) logging of all Traffic Manager transactions to the `lm.log` file.

Configuration Variable Data Type	Default Value	Description
proxy.config.admin.log_resolve_hostname INT	1	When enabled (1), the hostname of the client connecting to Traffic Manager is recorded in the `lm.log` file. When disabled (0), the IP address of the client connecting to Traffic Manager is not recorded in the `lm.log` file.
Process Manager		
proxy.config.process_manager.mgmt_port INT	8084	Specifies the port used for internal communication between the traffic_manager process and the traffic_server process.
Virtual IP Manager		
proxy.config.vmap.enabled INT	0	Enables (1) or disables (0) the Virtual IP option. Refer to *Using Virtual IP Failover, on page 155*.
Alarm Configuration		
proxy.config.alarm.bin STRING	example_alarm_bin.sh (UNIX) example_alarm.bat (Windows)	Specifies the name of the script file that can execute certain actions when an alarm is signaled. The default file is a sample script named `example_alarm_bin.sh` (UNIX) or `example_alarm_bin.bat` (Windows) located in the `bin` directory. You must edit the script to suit your needs.
proxy.config.alarm.abs_path STRING	NULL	Specifies the full path to the script file that sends email to alert someone of Traffic Edge problems.
ARM (Transparency Configuration)		
proxy.config.arm.enabled INT	0	Enables (1) or disables (0) the ARM, which is used for transparent proxy caching, IP spoofing, and ARM security. Refer to *Chapter 6, Transparent Proxy Caching* and *Chapter 14, Security Options*.
proxy.config.arm.ignore_ifp INT	1	Configures Traffic Edge to ignore the interface when sending packets back to the client if NAT rules are applied.

Configuration Variable Data Type	Default Value	Description
proxy.config.arm.always_query_dest INT	0	When enabled (1), Traffic Edge always asks the ARM driver for the original destination IP address of incoming requests. Traffic Edge then uses that IP address to determine the origin server rather than performing a DNS lookup on the hostname of the request. (Because the client already performed a DNS lookup, Traffic Edge does not have to.) Note: Inktomi recommends that you do not enable this variable if Traffic Edge is running in *both* explicit proxy caching mode and transparent proxy caching mode. In explicit proxy caching mode, the client does not perform a DNS lookup on the hostname of the origin server, so Traffic Edge must do it.
proxy.config.http.outgoing_ip_spoofing_enabled INT	0	Enables (1) or disables (0) the IP spoofing option, which allows Traffic Edge to establish connections to origin servers with the client IP address instead of Traffic Edge's IP address. Note: The variable proxy.config.arm.enabled must be enabled for the IP spoofing option to work.
proxy.config.arm.bypass_dynamic_enabled INT	0	Enables (1) or disables (0) the adaptive bypass option to bypass the proxy and go directly to the origin server when clients or servers cause problems.
proxy.config.arm.bypass_use_and_rules_bad_client_request INT	0	Enables (1) or disables (0) dynamic source/destination bypass in the event of nonHTTP traffic on port 80. Note: The variable proxy.config.arm.bypass_on_bad_client_request must also be enabled for this option to work.

Configuration Variable Data Type	Default Value	Description
proxy.config.arm.bypass_use_and_rules_400 INT	0	Enables (1) or disables (0) dynamic generation of source/ destination bypass rules when an origin server returns a 400 error. Note: The variable proxy.config.arm.bypass_on_400 must also be enabled for this option to work.
proxy.config.arm.bypass_use_and_rules_401 INT	0	Enables (1) or disables (0) dynamic generation of source/ destination bypass rules when an origin server returns a 401 error. Note: The variable proxy.config.arm.bypass_on_401 must also be enabled for this option to work.
proxy.config.arm.bypass_use_and_rules_403 INT	0	Enables (1) or disables (0) dynamic generation of source/ destination bypass rules when an origin server returns a 403 error. Note: The variable proxy.config.arm.bypass_on_403 must also be enabled for this option to work.
proxy.config.arm.bypass_use_and_rules_405 INT	0	Enables (1) or disables (0) dynamic generation of source/ destination bypass rules when an origin server returns a 405 error. Note: The variable proxy.config.arm.bypass_on_405 must also be enabled for this option to work.
proxy.config.arm.bypass_use_and_rules_406 INT	0	Enables (1) or disables (0) dynamic generation of source/ destination bypass rules when an origin server returns a 406 error. Note: The variable proxy.config.arm.bypass_on_406 must also be enabled for this option to work.

Configuration Variable Data Type	Default Value	Description
proxy.config.arm.bypass_use_and_rules_408 INT	0	Enables (1) or disables (0) dynamic generation of source/ destination bypass rules when an origin server returns a 408 error. Note: The variable proxy.config.arm.bypass_on_408 must also be enabled for this option to work.
proxy.config.arm.bypass_use_and_rules_500 INT	0	Enables (1) or disables (0) dynamic generation of source/ destination bypass rules when an origin server returns a 500 error. Note: The variable proxy.config.arm.bypass_on_500 must also be enabled for this option to work.
proxy.config.arm.bypass_on_bad_client_request INT	0	Enables (1) or disables (0) dynamic destination bypass in the event of nonHTTP traffic on port 80.
proxy.config.arm.bypass_on_400 INT	0	Enables (1) or disables (0) dynamic generation of destination bypass rules when an origin server returns a 400 error.
proxy.config.arm.bypass_on_401 INT	0	Enables (1) or disables (0) dynamic generation of destination bypass rules when an origin server returns a 401 error.
proxy.config.arm.bypass_on_403 INT	0	Enables (1) or disables (0) dynamic generation of destination bypass rules when an origin server returns a 403 error.
proxy.config.arm.bypass_on_405 INT	0	Enables (1) or disables (0) dynamic generation of destination bypass rules when an origin server returns a 405 error.
proxy.config.arm.bypass_on_406 INT	0	Enables (1) or disables (0) dynamic generation of destination bypass rules when an origin server returns a 406 error.

Configuration Variable Data Type	Default Value	Description
proxy.config.arm.bypass_on_408 INT	0	Enables (1) or disables (0) dynamic generation of destination bypass rules when an origin server returns a 408 error.
proxy.config.arm.bypass_on_500 INT	0	Enables (1) or disables (0) dynamic generation of destination bypass rules when an origin server returns a 500 error.
(ARM) Load Shedding Configuration		
proxy.config.arm.loadshedding.max_connections INT	1000000	Specifies the maximum number of client connections allowed before Traffic Edge starts forwarding incoming requests directly to the origin server.
Authentication Basic Realm		
proxy.config.proxy.authenticate.basic.realm	NULL	Specifies the authentication realm name. If the default of `NULL` is specified, `traffic_edge` is used. Important: For WMT (HTTP streaming), do not enter a value that contains the string `server`; otherwise, proxy authentication with the Windows Media Player does not work correctly.
LDAP		
proxy.config.ldap.auth.enabled INT	0	Enables (1) or disables (0) LDAP proxy authentication. Refer to *Using LDAP Proxy Authentication, on page 215* for information about configuring Traffic Edge to use LDAP proxy authentication.
proxy.config.ldap.auth.ttl_value INT	3000	Specifies the amount of time (in minutes) that entries in the Traffic Edge authentication cache remain valid.
proxy.config.ldap.auth.purge_cache_on_auth_fail INT	0	When enabled (1), configures Traffic Edge to delete the authorization entry for the client in the authentication cache if authentication fails.
proxy.config.ldap.proc.ldap.server.name STRING	NULL	Specifies the LDAP server name.

Configuration Variable Data Type	Default Value	Description
proxy.config.ldap.proc.ldap.server.port INT	389	Specifies the LDAP port.
proxy.config.ldap.proc.ldap.base.dn STRING	NULL	Specifies the LDAP Base Distinguished Name (DN). Obtain this value from your LDAP administrator.
proxy.config.ldap.proc.ldap.uid_filter	uid	Specifies the LDAP user ID. Use this as a filter to search the full DN database. Inktomi recommends that you do not modify this variable.
RADIUS Authentication		
proxy.config.radius.auth.enabled INT	0	Enables (1) or disables (0) RADIUS proxy authentication.
proxy.config.radius.proc.radius.primary_server.name STRING	NULL	Specifies the hostname or IP address of the primary RADIUS authentication server.
proxy.config.radius.proc.radius.primary_server.auth_port INT	1812	Specifies the port that Traffic Edge uses to communicate with the primary RADIUS authentication server.
proxy.config.radius.proc.radius.primary_server.shared_key STRING	NULL	Specifies the key used for encoding with the first RADIUS authentication server.
proxy.config.radius.proc.radius.secondary_server.name STRING	NULL	Specifies the hostname or IP address of the secondary RADIUS authentication server.
proxy.config.radius.proc.radius.secondary_server.auth_port INT	1812	Specifies the port that Traffic Edge uses to communicate with the secondary RADIUS authentication server.
proxy.config.radius.proc.radius.secondary_server.shared_key STRING	NULL	Specifies the key used for encoding with the secondary RADIUS authentication server.
proxy.config.radius.auth.min_timeout INT	10	Specifies the amount of time the Traffic Edge connection to the RADIUS server can remain idle before Traffic Edge closes the connection.
proxy.config.radius.auth.max_retries INT	10	Specifies the maximum number of times Traffic Edge can try to connect to the RADIUS server.

Configuration Variable Data Type	Default Value	Description
proxy.config.radius.auth.ttl_value INT	60	Specifies the number of minutes that Traffic Edge can store RADIUS username and password entries in the authentication cache. A value of 0 means that the entry never expires.
NTLM		
proxy.config.ntlm.auth.enabled INT	0	Enables (1) or disables (0) NTLM proxy authentication.
proxy.config.ntlm.dc.list STRING	NULL	Specifies the hostnames of the domain controllers. You must separate each entry with a comma: for example, host1, host2, host3.
proxy.config.ntlm.dc.load_balance INT	0	Enables (1) or disables (0) load balancing. When enabled, Traffic Edge balances the load when sending authentication requests to the domain controllers.
proxy.config.ntlm.dc.max_connections INT	3	Specifies the maximum number of connections Traffic Edge can have open to the domain controller.
proxy.config.ntlm.nt_domain STRING	NULL	Specifies the NT domain name against which Traffic Edge should authenticate.
proxy.config.ntlm.cache.ttl_value INT	3600	Specifies the number of seconds that Traffic Edge can store NTLM entries in the authentication cache.
HTTP Engine		
proxy.config.http.server_port INT	8080	Specifies the port that Traffic Edge uses when acting as a web proxy server for web traffic or when serving web traffic transparently.
proxy.config.http.server_port_attr STRING	X	Specifies the server port options. You can specify one of the following: C=SERVER_PORT_COMPRESSED X=SERVER_PORT_DEFAULT T=SERVER_PORT_BLIND_TUNNEL

Configuration Variable Data Type	Default Value	Description
proxy.config.http.server_other_ports STRING	NULL	Specifies the ports other than the port specified by the variable proxy.config.http.server_port to bind for incoming http requests.
proxy.config.http.ssl_ports STRING	443 563	Specifies the range of ports used for tunneling. Traffic Edge allows tunnels only to the specified ports; for example, to retrieve an object using HTTPS via Traffic Edge requires establishing a tunnel via Traffic Edge to an origin server.
proxy.config.http.insert_request_via_str INT	1	You can specify one of the following: 0 = no extra information is added to the string. 1 = all extra information is added. 2 = some extra information is added.
proxy.config.http.insert_response_via_str INT	1	You can specify one of the following: 0 = no extra information is added to the string. 1 = all extra information is added. 2 = some extra information is added.
proxy.config.http.enable_url_expandomatic INT	1	Enables (1) or disables (0) `.com` domain expansion, which configures the Traffic Edge to attempt to resolve unqualified hostnames by redirecting them to the expanded address, prepended with `www.` and appended with `.com`; for example, if a client makes a request to `host`, Traffic Edge redirects the request to `www.host.com`.
proxy.config.http.no_dns_just_forward_to_parent INT	0	When enabled (1), and if parent caching is enabled, Traffic Edge does no DNS lookups on request hostnames.

Configuration Variable Data Type	Default Value	Description
proxy.config.http.uncacheable_requests_bypass_parent INT	1	When enabled (1), Traffic Edge bypasses the parent proxy for a request that is not cacheable.
proxy.config.http.keep_alive_enabled INT	1	Enables (1) or disables (0) the use of keep-alive connections to either origin servers or clients.
proxy.config.http.chunking_enabled INT	0	Specifies whether Traffic Sever can generate a chunked response: 0 = Never 1 = Always 2 = Generate a chunked response if the server has returned HTTP 1.1 before 3 = Generate a chunked response if the client request is HTTP 1.1 and the origin server has returned HTTP 1.1 before
proxy.config.http.send_http11_requests INT	3	Configures Traffic Edge to use HTTP Version 1.1 when communicating with origin servers. You can specify one of the following values: 1 = Traffic Edge always uses HTTP 1.1 when communicating with origin servers. 2 = Traffic Edge uses HTTP 1.1 if the origin server has previously used HTTP 1.1. 3 = Traffic Edge uses HTTP 1.1 if the client request is HTTP 1.1 and the origin server has previously used HTTP 1.1. Note: If HTTP 1.1 is used, then Traffic Edge can use keep-alive connections with pipelining to origin servers. If HTTP 0.9 is used, then Traffic Edge does not use keep-alive connections to origin servers. If HTTP 1.0 is used, then a Traffic Edge can use keep-alive connections without pipelining to origin servers.

Configuration Variable Data Type	Default Value	Description
proxy.config.http.share_server_sessions INT	1	Enables (1) or disables (0) the re-use of server sessions.
proxy.config.http.ftp_enabled INT	1	Enables (1) or disables (0) Traffic Edge from serving FTP requests sent via HTTP.
proxy.config.http.record_heartbeat INT	0	Enables (1) or disables (0) traffic_cop heartbeat logging.
parent proxy configuration		
proxy.config.http.parent_proxy_routing_enable INT	0	Enables (1) or disables (0) the parent caching option. Refer to *Chapter 9, Hierarchical Caching*.
proxy.config.http.parent_proxy.retry_time INT	300	Specifies the amount of time allowed between connection retries to a parent cache that is unavailable.
proxy.config.http.parent_proxy.fail_threshold INT	10	Specifies the number of times the connection to the parent cache can fail before Traffic Edge considers the parent unavailable.
proxy.config.http.parent_proxy.total_connect_attempts INT	4	Specifies the total number of connection attempts allowed to a parent cache before Traffic Edge bypasses the parent or fails the request (depending on the go_direct option in the `bypass.config` file).
proxy.config.http.parent_proxy.per_parent_connect_attempts INT	2	Specifies the total number of connection attempts allowed per parent if multiple parents are used.
proxy.config.http.parent_proxy.connect_attempts_timeout INT	30	Specifies the timeout value in seconds for parent cache connection attempts.
proxy.config.http.forward.proxy_auth_to_parent INT	0	Configures Traffic Edge to send proxy authentication headers on to the parent cache.
HTTP connection timeouts (secs)		
proxy.config.http.keep_alive_no_activity_timeout_in INT	10	Specifies how long Traffic Edge keeps connections to clients open for a subsequent request after a transaction ends.

Configuration Variable Data Type	Default Value	Description
proxy.config.http.keep_alive_no_activity_timeout_out INT	10	Specifies how long Traffic Edge keeps connections to origin servers open for a subsequent transfer of data after a transaction ends.
proxy.config.http.transaction_no_activity_timeout_in INT	120	Specifies how long Traffic Edge keeps connections to clients open if a transaction stalls.
proxy.config.http.transaction_no_activity_timeout_out INT	120	Specifies how long Traffic Edge keeps connections to origin servers open if the transaction stalls.
proxy.config.http.transaction_active_timeout_in INT	0	Specifies the maximum amount of time Traffic Edge can remain connected to a client. If the transfer to the client is not complete before this timeout expires, Traffic Edge closes the connection. The default value of 0 specifies that there is no timeout.
proxy.config.http.transaction_active_timeout_out INT	0	Specifies the maximum amount of time Traffic Edge waits for fulfillment of a connection request to an origin server. If Traffic Edge does not complete the transfer to the origin server before this timeout expires, Traffic Edge terminates the connection request. The default value of 0 specifies that there is no timeout.
proxy.config.http.accept_no_activity_timeout INT	120	Specifies the timeout interval in seconds before Traffic Edge closes a connection that has no activity.
proxy.config.http.background_fill_active_timeout INT	60	Specifies how long Traffic Edge continues a background fill before giving up and dropping the origin server connection.

Configuration Variable Data Type	Default Value	Description
proxy.config.http.background_fill_completed_threshold FLOAT	0.50000	Specifies the proportion of total document size already transferred when a client aborts at which the proxy continues fetching the document from the origin server to get it into the cache (a *background fill*).
origin server connect attempts		
proxy.config.http.connect_attempts_max_retries INT	6	Specifies the maximum number of connection retries Traffic Edge can make when the origin server is not responding.
proxy.config.http.connect_attempts_max_retries_dead_server INT	2	Specifies the maximum number of connection retries Traffic Edge can make when the origin server is unavailable.
proxy.config.http.connect_attempts_rr_retries INT	2	Specifies the maximum number of failed connection attempts allowed before a round-robin entry is marked as down if a server has round-robin DNS entries.
proxy.config.http.connect_attempts_timeout INT	30	Specifies the timeout value in seconds for an origin server connection.
proxy.config.http.streaming_connect_attempts_timeout INT	1800	Specifies the timeout value in seconds for an origin server connection when the client request is from a streaming media client.
proxy.config.http.post_connect_attempts_timeout INT	1800	Specifies the timeout value in seconds for an origin server connection when the client request is a `POST` or `PUT` request.
proxy.config.http.down_server.cache_time INT	900	Specifies how long in seconds Traffic Edge remembers that an origin server was unreachable.
proxy.config.http.down_server.abort_threshold INT	10	Specifies the number of seconds before Traffic Edge marks an origin server as unavailable when a client abandons a request because the origin server was too slow in sending the response header.

Configuration Variable Data Type	Default Value	Description
congestion control		
proxy.config.http.congestion_control.enabled INT	0	Enables (1) or disables (0) the congestion control option, which configures Traffic Edge to stop forwarding HTTP requests to origin servers when they become congested. Traffic Edge sends the client a message to retry the congested origin server later. Refer to *Using Congestion Control, on page 58*.
negative response caching		
proxy.config.http.negative_caching_enabled INT	0	When enabled (1), Traffic Edge caches negative responses, such as *404 Not Found,* if a requested page does not exist. The next time a client requests the same page, Traffic Edge serves the negative response from the cache. The following lists the negative responses that Traffic Edge caches: 204 No Content 305 Use Proxy 400 Bad Request 403 Forbidden 404 Not Found 405 Method Not Allowed 500 Internal Server Error 501 Not Implemented 502 Bad Gateway 503 Service Unavailable 504 Gateway Timeout
proxy.config.http.negative_caching_lifetime INT	1800	Specifies the how long Traffic Edge keeps the negative responses as valid in cache.
proxy users variables		
proxy.config.http.anonymize_remove_from INT	0	When enabled (1), Traffic Edge removes the `From` header that accompanies transactions to protect the privacy of your users.

Configuration Variable Data Type	Default Value	Description
proxy.config.http.anonymize_remove_referer INT	0	When enabled (1), Traffic Edge removes the `Referer` header that accompanies transactions to protect the privacy of your site and users.
proxy.config.http.anonymize_remove_user_agent INT	0	When enabled (1), Traffic Edge removes the `User-agent` header that accompanies transactions to protect the privacy of your site and users.
proxy.config.http.anonymize_remove_cookie INT	0	When enabled (1), Traffic Edge removes the `Cookie` header that accompanies transactions to protect the privacy of your site and users.
proxy.config.http.anonymize_remove_client_ip INT	0	When enabled (1), Traffic Edge removes `Client-IP` headers for more privacy.
proxy.config.http.anonymize_insert_client_ip INT	1	When enabled (1), Traffic Edge inserts `Client-IP` headers to retain the client IP address.
proxy.config.http.append_xforwards_header INT	0	When enabled (1), Traffic Edge appends `X-Forwards` headers to outgoing requests.
proxy.config.http.anonymize_other_header_list STRING	NULL	Specifies the headers that Traffic Edge will remove from outgoing requests.
proxy.config.http.snarf_username_from_authorization INT	0	When enabled (1), Traffic Edge takes the username and password from the authorization header for LDAP if the authorization scheme is *Basic*.
proxy.config.http.insert_squid_x_forwarded_for INT	0	When enabled (1), Traffic Edge adds the client IP address to the `X-Forwarded-For` header.

Configuration Variable Data Type	Default Value	Description
security		
proxy.config.http.push_method_enabled INT	0	Enables (1) or disables (0) the HTTP PUSH option, which allows you to deliver content directly to the cache without user request. Important: If you enable this option, you must also specify a filtering rule in the `filter.config` file to allow only certain machines to push content into the cache. Refer to *filter.config, on page 375*.
cache control		
proxy.config.http.cache.http INT	1	Enables (1) or disables (0) caching of HTTP requests.
proxy.config.http.cache.ftp INT	1	Enables (1) or disables (0) caching of FTP requests sent via HTTP.
proxy.config.http.cache.ignore_client_no_cache INT	0	When enabled (1), Traffic Edge ignores client requests to bypass the cache.
proxy.config.http.cache.ims_on_client_no_cache INT	0	When enabled (1), Traffic Edge issues a conditional request to the origin server if an incoming request has a `no-cache` header.
proxy.config.http.cache.ignore_server_no_cache INT	0	When enabled (1), Traffic Edge ignores origin server requests to bypass the cache.
proxy.config.http.cache.cache_responses_to_cookies INT	3	Specifies how cookies are cached: 0 = do not cache any responses to cookies 1 = cache for any content-type 2 = cache only for image types 3 = cache for all but text content-types
proxy.config.http.cache.ignore_authentication INT	0	When enabled (1), Traffic Edge ignores `WWW-Authentication` headers in responses. `WWW-Authentication` headers are removed and not cached.
proxy.config.http.cache.cache_urls_that_look_dynamic INT	0	Enables (1) or disables (0) caching of URLs that look dynamic.

Configuration Variable Data Type	Default Value	Description
proxy.config.http.cache.enable_default_vary_headers INT	0	Enables (1) or disables (0) caching of alternate versions of HTTP objects that do not contain the Vary header.
proxy.config.http.cache.when_to_revalidate INT	0	Specifies when to revalidate content: 0 = Use cache directives or heuristic (the default value). 1 = Stale if heuristic. 2 = Always stale (always revalidate). 3 = Never stale. 4 = Use cache directives or heuristic (0) unless the request has an If-Modified-Since header. If the request has an If-Modified-Since header, Traffic Edge always revalidates the cached content and uses the client's If-Modified-Since header for the proxy request.
proxy.config.http.cache.when_to_add_no_cache_to_msie_requests INT	0	Specifies when to add `no-cache` directives to Microsoft Internet Explorer requests. You can specify the following: 0 = `no-cache` not added to MSIE requests. 1 = `no-cache` added to IMS MSIE requests. 2 = `no-cache` added to all MSIE requests.
proxy.config.http.cache.required_headers INT	0	Specifies the type of headers required in a request for the request to be cacheable. 0 = no required headers to make document cacheable. 1 = at least `Last-Modified` header required. 2 = explicit lifetime required, `Expires` or `Cache-Control`.
proxy.config.http.cache.max_stale_age INT	604800	Specifies the maximum age allowed for a stale response before it cannot be cached.
proxy.config.http.cache.range.lookup INT	1	When enabled (1), Traffic Edge looks up range requests in the cache.

Configuration Variable Data Type	Default Value	Description
heuristic expiration		
proxy.config.http.cache.heuristic_min_lifetime INT	3600	Specifies the minimum amount of time that a document in the cache can be considered fresh.
proxy.config.http.cache.heuristic_max_lifetime INT	86400	Specifies the maximum amount of time that a document in the cache can be considered fresh.
proxy.config.http.cache.heuristic_lm_factor FLOAT	0.10000	Specifies the aging factor for freshness computations.
proxy.config.http.cache.fuzz.time INT	240	Specifies the interval in seconds before the document stale time that Traffic Edge checks for an early refresh.
proxy.config.http.cache.fuzz.probability FLOAT	0.00500	Specifies the probability that a refresh is made on a document during the specified fuzz time.
dynamic content & content negotiation		
proxy.config.http.cache.vary_default_text STRING	NULL	Specifies the header on which Traffic Edge varies for text documents; for example, if you specify `user-agent`, Traffic Edge caches all the different user-agent versions of documents it encounters.
proxy.config.http.cache.vary_default_images STRING	NULL	Specifies the header on which Traffic Edge varies for images.
proxy.config.http.cache.vary_default_other STRING	NULL	Specifies the header on which Traffic Edge varies for anything other than text and images.
anonymous ftp password		
proxy.config.http.ftp.anonymous_passwd STRING	inktomi	Specifies the anonymous password for FTP servers that require a password for access. Traffic Edge uses the Traffic Edge user account name as the default value for this variable.
cached ftp document lifetime		
proxy.config.http.ftp.cache.document_lifetime INT	259200	Specifies the maximum amount of time that an FTP document can stay in the Traffic Edge cache.

Configuration Variable Data Type	Default Value	Description
ftp transfer mode		
proxy.config.http.ftp.binary_transfer_only INT	0	When enabled (1), all FTP documents requested from HTTP clients are transferred in binary mode only. When disabled (0), FTP documents requested from HTTP clients are transferred in ASCII or binary mode, depending on the document type.
Customizable User Response Pages		
proxy.config.body_factory.enable_customizations INT	0	Specifies whether customizable response pages are enabled or disabled and which response pages are used: 0 = disable customizable user response pages 1 = enable customizable user response pages in the default directory only 2 = enable language-targeted user response pages
proxy.config.body_factory.enable_logging INT	1	Enables (1) or disables (0) logging for customizable response pages. When enabled, Traffic Edge records a message in the error log each time a customized response page is used or modified.
proxy.config.body_factory.template_sets_dir	config/body_factory	Specifies the customizable response page default directory.
proxy.config.body_factory.response_suppression_mode INT	0	Specifies when Traffic Edge suppresses generated response pages: 0 = never suppress generated response pages 1 = always suppress generated response pages 2 = suppress response pages only for intercepted traffic

Configuration Variable Data Type	Default Value	Description
FTP Engine		
Ftp over Http		
proxy.config.ftp.data_connection_mode INT	1	Specifies the FTP connection mode: 1 = PASV then PORT 2 = PORT only 3 = PASV only
proxy.config.ftp.control_connection_timeout INT	300	Specifies how long Traffic Edge waits for a response from the FTP server.
proxy.config.ftp.rc_to_switch_to_PORT STRING	NULL	Specifies the response codes for which Traffic Edge automatically fails over to the PORT command when PASV fails if the configuration variable proxy.config.ftp.data_connection_mode is set to 1. This variable is used for FTP requests from HTTP clients only.
Ftp Proxy		
proxy.config.ftp.ftp_enabled INT	0	Enables (1) or disables (0) processing of FTP requests from FTP clients.
proxy.config.ftp.cache_enabled INT	1	Enables (1) or disables (0) FTP documents to be put in the cache. If this option is disabled, Traffic Edge always serves FTP documents from the FTP server.
proxy.config.ftp.logging_enabled INT	1	Enables (1) or disables (0) logging of FTP transactions.
proxy.config.ftp.proxy_server_port INT	21	Specifies the port used for FTP connections.

Configuration Variable Data Type	Default Value	Description
proxy.config.ftp.open_lisn_port_mode INT	1	Specifies how FTP opens a listening port for a data transfer: 1 = The operating system chooses an available port. Traffic Edge sends 0 and retrieves the new port number if the listen succeeds. 2 = The listening port is determined by the range of ports specified by the Traffic Edge variables proxy.config.ftp.min_lisn_port and proxy.config.ftp.max_lisn_port, described below.
proxy.config.ftp.min_lisn_port INT	32768	Specifies the lowest port in the range of listening ports used by Traffic Edge for data connections when the FTP client sends a PASV or Traffic Edge sends a PORT to the FTP server.
proxy.config.ftp.max_lisn_port INT	65535	Specifies the highest port in the range of listening ports used by Traffic Edge for data connections when the FTP client sends a PASV or Traffic Edge sends a PORT to the FTP server.
proxy.config.ftp.server_data_default_pasv INT	1	Specifies the default method used to set up server side data connections: 1 = Traffic Edge sends a PASV to the FTP server and lets the FTP server open a listening port. 0 = Traffic Edge tries PORT first (sets up a listening port on the Traffic Edge side of the connection).

Configuration Variable Data Type	Default Value	Description
proxy.config.ftp.different_client_port_ip_allowed INT	0	When enabled (1), Traffic Edge can connect to a machine other than the one on which the FTP client is running to establish a data connection. The FTP client uses PORT to set up a listening port on its side and allows Traffic Edge to connect to that port to establish the data connection (used to transfer files). When setting up the listening port, an FTP client specifies the IP address and port number for the listening port. If this variable is set to 0 (zero), Traffic Edge cannot connect to the FTP client if the IP address sent by the client is different from the IP address of the machine running the FTP client.
proxy.config.ftp.try_pasv_times INT	1024	Specifies the number of times Traffic Edge can try to open a listening port when the FTP client sends a PASV.
proxy.config.ftp.try_port_times INT	1024	Specifies the maximum number of times Traffic Edge can try to open a listening port when sending a PORT to the FTP server.
proxy.config.ftp.try_server_ctrl_connect_times INT	6	Specifies the maximum number of times Traffic Edge can try to connect to the FTP server's control listening port.
proxy.config.ftp.try_server_data_connect_times INT	3	Specifies the maximum number of times Traffic Edge can try to connect to the FTP server's data listening port when it sends a PASV to the FTP server and gets the IP/ listening port information.
proxy.config.ftp.try_client_data_connect_times INT	3	Specifies the maximum number of times Traffic Edge can try to connect to the FTP client's data listening port when the FTP client sends a PORT with the IP/listening port information.

Configuration Variable Data Type	Default Value	Description
proxy.config.ftp.client_ctrl_no_activity_timeout INT	900	Specifies the no activity timeout for the FTP client control connection.
proxy.config.ftp.client_ctrl_active_timeout INT	14400	Specifies the active timeout for the FTP client control connection.
proxy.config.ftp.server_ctrl_no_activity_timeout INT	120	Specifies the inactivity timeout for the FTP server control connection.
proxy.config.ftp.server_ctrl_active_timeout INT	14400	Specifies the active timeout for the FTP server control connection.
proxy.config.ftp.client_data_no_activity_timeout INT	120	Specifies the inactivity timeout for the FTP data connection on the client side.
proxy.config.ftp.client_data_active_timeout INT	14400	Specifies the active timeout for the FTP data connection on the client side.
proxy.config.ftp.server_data_no_activity_timeout INT	120	Specifies the inactivity timeout for the FTP data connection on the server side.
proxy.config.ftp.server_data_active_timeout INT	14400	Specifies the active timeout for the FTP data connection on the server side.
proxy.config.ftp.pasv_accept_timeout INT	120	Specifies the timeout value for a listening data port in Traffic Edge (for PASV, the client data connection).
proxy.config.ftp.port_accept_timeout INT	120	Specifies the timeout value for a listening data port in Traffic Edge (for PORT, the server data connection).
proxy.config.ftp.share_ftp_server_ctrl_enabled INT	1	Enables (1) or disables (0) sharing the server control connections among multiple anonymous FTP clients.

Configuration Variable Data Type	Default Value	Description
proxy.config.ftp.share_only_after_session_end INT	1	Specifies how an FTP server control connection is shared between different FTP client sessions: 1 = the FTP server control connection can be used by another FTP client session *only* when the FTP client session is complete (typically, when the FTP client sends out a QUIT command). 0 = the FTP server control connection can be used by another FTP client session *only* if the FTP client session is not actively using the FTP server connection: for example, if the request is a cache hit or during an idle session.
proxy.config.ftp.server_ctrl_keep_alive_no_activity_timeout INT	90	Specifies the timeout value when the FTP server control connection is not used by any FTP clients.
proxy.config.ftp.reverse_ftp_enabled INT	0	Enables (1) or disables (0) the FTP reverse proxy option. If enabled, you must configure the ftp_remap.config file. Refer to *Setting FTP Mapping Rules, on page 139*.
proxy.config.ftp.login_info_fresh_in_cache_time INT	604800	Specifies how long the 220/230 responses (login messages) can stay fresh in the cache.
proxy.config.ftp.directory_listing_fresh_in_cache_time INT	86400	Specifies how long directory listings can stay fresh in the cache.
proxy.config.ftp.file_fresh_in_cache_time INT	259200	Specifies how long FTP files can stay fresh in the cache.
proxy.config.ftp.simple_directory_listing_cache_enabled INT	1	Enables (1) or disables (0) caching of directory listings without arguments: for example, `dir/ls`.
proxy.config.ftp.full_directory_listing_cache_enabled INT	1	Enables (1) or disables (0) caching of directory listings with arguments: for example, `ls -al, ls *.txt`.

Configuration Variable Data Type	Default Value	Description
proxy.config.ftp.file_fresh_mdtm_checking_enabled INT	0	Enables (1) or disables (0) the MDTM extension command to check the freshness of the cached document (if the command is supported on the server).
proxy.config.ftp.data_source_port_20_enabled INT	0	When enabled (1) and Traffic Edge connects to a client to create a new data connection, the source port for the data connection is 20 instead of a random number.
SOCKS Processor		
proxy.config.socks.socks_needed INT	0	Enables (1) or disables (0) the SOCKS option. Refer to *Configuring SOCKS Firewall Integration, on page 210*.
proxy.config.socks.socks_version INT	4	Specifies the SOCKS version.
proxy.config.socks.default_servers STRING		Specifies the names and ports of the SOCKS servers with which Traffic Edge communicates.
proxy.config.socks.accept_enabled INT	0	Enables (1) or disables (0) the SOCKS proxy option. As a SOCKS proxy, Traffic Edge receives SOCKS traffic (usually on port 1080), detects and serves HTTP requests and forwards all other requests directly to the SOCKS server.
proxy.config.socks.accept_port INT	1080	Specifies the port on which Traffic Edge accepts SOCKS traffic.
Net Subsystem		
proxy.config.net.connections_throttle INT	10000	Specifies the maximum number of connections that Traffic Edge can handle. If Traffic Edge receives additional client requests, they are queued until existing requests are served. Important: Do not set this variable below the minimum value of 100.

Configuration Variable Data Type	Default Value	Description
Cluster Subsystem		
proxy.config.cluster.cluster_port INT	8086	Specifies the port used for cluster communication.
proxy.config.cluster.ethernet_interface STRING	*your_interface*	Specifies the network interface used for cluster traffic. All nodes in a cluster must use the same network interface.
Cache		
proxy.config.cache.permit.pinning INT	0	Enables (1) or disables (0) the cache pinning option, which lets you keep objects in the cache for a specified time. You set cache pinning rules in the `cache.config` file (refer to *cache.config, on page 369*).
proxy.config.cache.ram_cache.size INT	-1	Specifies the size of the RAM cache, in bytes. -1 means that the RAM cache is automatically sized at approximately one MB per gigabyte of disk.
proxy.config.cache.limits.http.max_alts INT	3	Specifies the maximum number of HTTP alternates that Traffic Edge can cache.
proxy.config.cache.max_doc_size INT	0	Specifies the maximum size of documents in the cache (in bytes): 0 = there is no size limit.
DNS		
proxy.config.dns.search_default_domains INT	1	Enables (1) or disables (0) local domain expansion so that Traffic Edge can attempt to resolve unqualified hostnames by expanding to the local domain; for example, if a client makes a request to an unqualified host named `host_x`, and if the Traffic Edge local domain is `y.com`, the Traffic Edge will expand the hostname to `host_x.y.com`.

Configuration Variable Data Type	Default Value	Description
proxy.config.dns.splitDNS.enabled INT	0	Enables (1) or disables (0) DNS server selection. When enabled, Traffic Edge refers to the `splitdns.config` file for the selection specification. Refer to *Configuring DNS Server Selection (Split DNS), on page 213*.
proxy.config.dns.splitdns.def_domain STRING	NULL	Specifies the default domain for split DNS requests. This value is appended automatically to the hostname if it does not include a domain before split DNS determines which DNS server to use.
proxy.config.dns.url_expansions STRING	NULL	Specifies a list of hostname extensions that are automatically added to the hostname after a failed lookup; for example, if you want Traffic Edge to add the hostname extension .org, specify `org` as the value for this variable (Traffic Edge automatically adds the dot (.). Note: If the variable `proxy.config.http.enable_url_expandomatic` is set to 1 (the default value), you do not have to add www. and .com to this list; Traffic Edge tries www. and .com automatically after trying the values you specify.
proxy.config.dns.round_robin_nameservers INT	0	Enables (1) or disables (0) DNS server round-robin.
proxy.config.dns.nameservers STRING	NULL	Specifies the DNS servers.
DNS Proxy		
proxy.config.dns.proxy.enabled INT	0	Enables (1) or disables (0) the DNS proxy caching option that lets you resolve DNS requests on behalf of clients. This option offloads remote DNS servers and reduces response time for DNS lookups. Refer to *Chapter 11, DNS Proxy Caching*.
proxy.config.dns.proxy_port INT	53	Specifies the port that Traffic Edge uses for DNS traffic.

Configuration Variable Data Type	Default Value	Description
HostDB		
proxy.config.hostdb.size INT	200000	Specifies the maximum number of entries allowed in the host database.
proxy.config.hostdb.ttl_mode INT	0	Specifies the host database time to live mode. You can specify one of the following: 0 = obey 1 = ignore 2 = min(X,ttl) 3 = max(X,ttl)
proxy.config.hostdb.timeout INT	1440	Specifies the foreground timeout, in minutes.
proxy.config.hostdb.strict_round_robin INT	0	When disabled (0), Traffic Edge always uses the same origin server for the same client as long as the origin server is available.
Logging Config		
proxy.config.log2.logging_enabled INT	3	Enables and disables event logging: 0 = logging disabled 1 = log errors only 2 = log transactions only 3 = full logging (errors + transactions) Refer to *Chapter 15, Working with Log Files*.
proxy.config.log2.max_secs_per_buffer INT	5	Specifies the maximum amount of time before data in the buffer is flushed to disk.
proxy.config.log2.max_space_mb_for_logs INT	2000	Specifies the amount of space allocated to the logging directory, in megabytes.
proxy.config.log2.max_space_mb_for_orphan_logs INT	25	Specifies the amount of space allocated to the logging directory, in megabytes if this node is acting as a collation client.

Configuration Variable Data Type	Default Value	Description
proxy.config.log2.max_space_mb_headroom INT	10	Specifies the tolerance for the log space limit in bytes. If the variable proxy.config.log2.auto_delete_rolled_file is set to 1 (enabled), autodeletion of log files is triggered when the amount of free space available in the logging directory is less than the value specified here.
proxy.config.log2.hostname STRING	localhost	Specifies the hostname of the machine running Traffic Edge.
proxy.config.log2.logfile_dir STRING	*install_dir*/logs	Specifies the full path to the logging directory.
proxy.config.log2.logfile_perm STRING	rw-r--r--	Specifies the log file permissions. The standard UNIX file permissions are used (owner, group, other). Valid values are: - no permission r read permission w write permission x execute permission Permissions are subject to the umask settings for the Traffic Edge process. This means that a umask setting of 002 will not allow write permission for others, even if specified in the configuration file. Permissions for existing log files are not changed when the configuration is changed. UNIX only.
proxy.config.log2.custom_logs_enabled INT	0	Enables (1) or disables (0) custom logging.

Configuration Variable Data Type	Default Value	Description
proxy.config.log2.xml_logs_config INT	1	Enables (1) or disables (0) extended custom logging using an XLM-based configuration file. A value of 0 instructs Traffic Edge to use the traditional custom log formats. Note: Previous Traffic Edge releases use the traditional custom logging option in addition to the XML-based custom logging option. Although this release of Traffic Edge continues to support traditional custom logging, Inktomi recommends that you use the XML-based custom formats, which are more versatile.
proxy.config.log2.squid_log_enabled INT	1	Enables (1) or disables (0) the squid log file format.
proxy.config.log2.squid_log_is_ascii INT	1	Specifies the squid log file type: 1 = ASCII 0 = binary
proxy.config.log2.squid_log_name STRING	squid	Specifies the squid log filename.
proxy.config.log2.squid_log_header STRING	NULL	Specifies the squid log file header text.
proxy.config.log2.common_log_enabled INT	0	Enables (1) or disables (0) the Netscape common log file format.
proxy.config.log2.common_log_is_ascii INT	1	Specifies the Netscape common log file type: 1 = ASCII 0 = binary
proxy.config.log2.common_log_name STRING	common	Specifies the Netscape common log filename.
proxy.config.log2.common_log_header STRING	NULL	Specifies the Netscape common log file header text.
proxy.config.log2.extended_log_enabled INT	0	Enables (1) or disables (0) the Netscape extended log file format.
proxy.confg.log2.extended_log_is_ascii INT	1	Specifies the Netscape extended log file type: 1 = ASCII 0 = binary

Configuration Variable Data Type	Default Value	Description
proxy.config.log2.extended_log_name STRING	extended	Specifies the Netscape extended log filename.
proxy.config.log2.extended_log_header STRING	NULL	Specifies the Netscape extended log file header text.
proxy.config.log2.extended2_log_enabled INT	0	Enables (1) or disables (0) the Netscape Extended-2 log file format.
proxy.config.log2.extended2_log_is_ascii INT	1	Specifies the Netscape Extended-2 log file type: 1 = ASCII 0 = binary
proxy.config.log2.extended2_log_name STRING	extended2	Specifies the Netscape Extended-2 log filename.
proxy.config.log2.extended2_log_header STRING	NULL	Specifies the Netscape Extended-2 log file header text.
proxy.config.log2.separate_icp_logs INT	0	When enabled (1), configures Traffic Edge to store ICP transactions in a separate log file.
proxy.config.log2.separate_mixt_logs INT	-1	When enabled (1), Traffic Edge records streaming media transactions in a separate standard log file: for example, `squid-mixt.log`. When disabled (0), Traffic Edge records streaming media transactions in the same standard log file as nonstreaming transactions: for example, `squid.log`. The default -1 configures Traffic Edge to create one standard log file without streaming media transactions.
proxy.config.log2.separate_host_logs INT	0	When enabled (1), configures Traffic Edge to create a separate log file for HTTP/FTP transactions for each origin server listed in the `log_hosts.config` file (refer to *HTTP Host Log Splitting, on page 255*).

Configuration Variable Data Type	Default Value	Description
proxy.local.log2.collation_mode INT	0	Specifies the log collation mode: 0 = Collation disabled. 1 = This host is a log collation server. 2 = This host is a collation client and sends entries using standard formats to the collation server. 3 = This host is a collation client and sends entries using the traditional custom formats to the collation server. 4 = This host is a collation client and sends entries that use both the standard and traditional custom formats to the collation server. For information on sending XML-based custom formats to the collation server, refer to *logs_xml.config, on page 387*. Note: Previous Traffic Edge releases use the traditional custom logging option in addition to the XML-based custom logging option. Although this release of Traffic Edge continues to support traditional custom logging, Inktomi recommends that you use the XML-based custom formats, which are more versatile.
proxy.confg.log2.collation_host STRING	NULL	Specifies the hostname of the log collation server.
proxy.config.log2.collation_port INT	8085	Specifies the port used for communication between the collation server and client.
proxy.config.log2.collation_secret STRING	foobar	Specifies the password used to validate logging data and prevent the exchange of unauthorized information when a collation server is being used.
proxy.config.log2.collation_host_tagged INT	0	When enabled (1), configures Traffic Edge to include the hostname of the collation client that generated the log entry in each entry.

Configuration Variable Data Type	Default Value	Description
proxy.config.log2.collation_retry_sec INT	5	Specifies the number of seconds between collation server connection retries.
proxy.config.log2.rolling_enabled INT	1	Specifies how log files are rolled. You can specify the following values: 0 to disable log file rolling. 1 to enable log file rolling at specific intervals during the day (specified with the proxy.config.log2.rolling_interval_sec and proxy.config.log2.rolling_offset_hr variables). 2 to enable log file rolling when log files reach a specific size (specified with the proxy.config.log2.rolling_size_mb variable). 3 to enable log file rolling at specific intervals during the day or when log files reach a specific size (whichever occurs first). 4 to enable log file rolling at specific intervals during the day when log files reach a specific size (at a specified time if the file is of the specified size). Refer to *Rolling Event Log Files, on page 252*.
proxy.config.log2.rolling_interval_sec INT	86400	Specifies the log file rolling interval, in seconds. The minimum value is 300 (5 minutes). The maximum value is 86400 seconds (one day).
proxy.config.log2.rolling_offset_hr INT	0	Specifies the file rolling offset hour. The hour of the day that starts the log rolling period.
proxy.config.log2.rolling_size_mb INT	10	Specifies the size that log files must reach before rolling takes place.
proxy.config.log2.auto_delete_rolled_files INT	1	Enables (1) or disables (0) automatic deletion of rolled files.

Configuration Variable Data Type	Default Value	Description
proxy.config.log2.sampling_frequency INT	1	Configures Traffic Edge to log only a sample of transactions rather than every transaction. You can specify the following values: 1 = log every transaction 2 = log every second transaction 3 = log every third transaction and so on...
Tunable for MIXT in general		
proxy.config.cache.ram_cache_mixt_cutoff INT	1048576	Specifies the maximum size of streams allowed in the RAM cache, in bytes.
proxy.config.resource.target_maxmem_mb INT	1382	Specifies the maximum memory usage allowed before all new connections from QuickTime and WMT are denied. This variable does not affect Real media player connections.
Windows Media Technology Configuration		
proxy.config.wmt.enabled INT	1	Enables (1) or disables (0) WMT caching and WMT options in Traffic Manager.
proxy.config.wmt.port INT	1755	Specifies the TCP and UDP port on which Traffic Edge listens for MMS protocol requests.
proxy.config.wmt.asx_rewrite.enabled INT	1	Enables (1) or disables (0) WMT `.ASX` file rewriting.
proxy.config.wmt.media_bridge.name STRING	NULL	Specifies the name of the MediaBridge node.
proxy.config.wmt.media_bridge.port INT	10022	Specifies the port of the MediaBridge node.
proxy.config.wmt.media_bridge.mount_point STRING	ffnet	Specifies the mount point for MDN streams. This is similar to the alias in the unicast publishing points of WMT servers.
proxy.config.wmt.media_bridge.monitor.version INT	0	Enables (1) or disables (0) the WMT ServerLinks. The value `3` enables WMT ServerLink monitoring.
proxy.config.wmt.media_bridge.monitor.port INT	10088	Specifies the port of the monitoring agent.

Configuration Variable Data Type	Default Value	Description
proxy.config.wmt.media_bridge.monitor.name STRING	NULL	Specifies the name of the monitoring agent.
proxy.config.wmt.max_rexmit_memory INT	20971520	Specifies the amount of memory Traffic Edge uses to store data to reply to retransmission requests.
proxy.config.wmt.inactivity_timeout INT	21600	Specifies how long a connection between Traffic Edge and the WMT server can remain inactive before Traffic Edge closes the connection.
MIXT WMTMCast Configuration		
proxy.config.mixt.wmtmcast.enabled INT	1	Enables (1) or disables (0) the WMT multicast option.
MIXT Push Configuration		
proxy.config.mixt.push.enabled INT	1	Enables (1) or disables (0) WMT media push.
proxy.config.mixt.push.port INT	1900	Specifies the port used for WMT media push. The value can be any valid port number.
proxy.config.mixt.push.password STRING	NULL	Specifies the password that must be given to preload files into the cache using the media push option. The value `NULL` disables password authentication. You can enter any ASCII string. Spaces are not allowed.
QuickTime Config		
proxy.config.qt.enabled INT	1	Enables (1) or disables (0) processing of QuickTime requests.
proxy.config.mixt.rtsp_proxy_port INT	554	Specifies the port Traffic Edge uses for all QuickTime requests and all transparent and reverse proxy Real media player requests.
proxy.config.qt.media_bridge.name STRING	NULL	Specifies the name of the MediaBridge node.
proxy.config.qt.media_bridge.port INT	10036	Specifies the port of the MediaBridge node.
proxy.config.qt.media_bridge.mount_point STRING	ffnet	Specifies the mount point for MDN streams.
proxy.config.qt.media_bridge.monitor.name STRING	NULL	Specifies the name of the monitoring agent.

Configuration Variable Data Type	Default Value	Description
proxy.config.qt.media_bridge.monitor.port INT	10088	Specifies the port of the monitoring agent.
RNI Config		
proxy.config.rni.enabled INT	1	Enables (1) or disables (0) processing of Real Networks requests.
proxy.config.rni.watcher_enabled INT	0	When enabled (1), the `traffic_cop` process starts the Real Proxy.
proxy.config.rni.proxy_port INT	9231	Specifies the port that Traffic Edge uses to contact RealProxy to serve explicit proxy requests from Real media players. If the QuickTime option is installed, the default value is 9231; otherwise, the default value is 554.
proxy.config.rni.proxy_pid_path STRING	NULL	Specifies the path to the RealProxy PID file. The value is generated automatically by RealProxy during startup and is used by the `traffic_cop` process to ensure that RealProxy is running. Important: Do not change the value of this variable.
proxy.config.rni.proxy_restart_cmd STRING	NULL	Specifies the exact command line (including the pathname) that Traffic Edge uses to start RealProxy. Important: Do not change the value of this variable.
proxy.config.rni.proxy_restart_interval INT	10	Specifies the amount of time that the `traffic_cop` process waits before restarting RealProxy.
proxy.config.rni.proxy_service_name STRING	RMProxy	Specifies the name of the RealProxy service.
proxy.config.rni.auth_port INT	7808	Specifies the port used to support LDAP and NTLM proxy authentication for Real Networks streams. Important: Do not change the value of this variable.
proxy.config.rni.rpass_watcher_enabled INT	0	When enabled (1), the `traffic_cop` process starts the Real Networks passthrough daemon.

Configuration Variable Data Type	Default Value	Description
proxy.config.rni.rpass_restart_cmd STRING	/export/home/ inktomi/5.2.2/bin	Specifies the exact command line (including the pathname) that must be used by the `start_traffic_server` command to execute the Real Networks passthrough proxy (rtspd).
Reverse Proxy		
proxy.config.reverse_proxy.enabled INT	0	Enables (1) or disables (0) HTTP and streaming media reverse proxy. For FTP reverse proxy, refer to *proxy.config.ftp.reverse_ftp_enabled, on page 424*.
proxy.config.header.parse.no_host_url_redirect STRING	NULL	Specifies the URL to which to redirect requests with no host headers (reverse proxy).
URL Remap Rules		
proxy.config.url_remap.default_to_server_pac INT	0	Enables (1) or disables (0) requests for a PAC file on the proxy service port (8080 by default) to be redirected to the PAC port. For this type of redirection to work, the variable proxy.config.reverse_proxy.enabled must be set to 1.

Configuration Variable Data Type	Default Value	Description
proxy.config.url_remap.default_to_server_pac_port INT	-1	Sets the PAC port so that PAC requests made to the Traffic Edge proxy service port are redirected this port. -1 specifies that the PAC port will be set to the autoconfiguration port (the default autoconfiguration port is 8083). This is the default setting. This variable can be used together with the proxy.config.url_remap.default_to_server_pac variable to get a PAC file from a different port. You must create and run a process that serves a PAC file on this port; for example, if you create a Perl script that listens on port 9000 and writes a PAC file in response to any request, you can set this variable to 9000 and browsers that request the PAC file from a proxy server on port 8080, will get the PAC file served by the Perl script.
proxy.config.url_remap.remap_required INT	0	Set this variable to 1 if you want Traffic Edge to serve requests only from origin servers listed in the mapping rules of the remap.config file. If a request does not match, the browser will receive an error.
proxy.config.url_remap.pristine_host_hdr INT	0	Set this variable to 1 if you want to retain the client host header in a request during remapping.
SSL Termination		
proxy.config.ssl.enabled INT	0	Enables (1) or disables (0) the SSL termination option. Refer to *Using SSL Termination, on page 226*.
proxy.config.ssl.SSLv2 INT	1	Enables (1) or disables (0) SSLv2.
proxy.config.ssl.SSLv3 INT	1	Enables (1) or disables (0) SSLv3.

Configuration Variable Data Type	Default Value	Description
proxy.config.ssl.TLSv1 INT	1	Enables (1) or disables (0) TLSv1. Note: IBM web servers do not support the TLS (Transport Layer Security) protocol. For IBM web servers to work with Traffic Edge, you must set this variable to 0.
proxy.confg.ssl.accelerator.type INT	0	Specifies the type of SSL accelerator card installed on your Traffic Edge machine: 0 = none (no SSL accelerator card is installed on the Traffic Edge machine. The CPU on the Traffic Edge machine determines the number of requests served per second). 1 = nCipher nFast accelerator card 2 = Rainbow CryptoSwift accelerator card 3 = Compaq Atalla accelerator card
proxy.config.ssl.atalla.lib.path STRING	/opt/atalla/lib	Specifies the library path for the Compaq Atalla accelerator card. You need only change this variable if you did not use the default path when you installed the card.
proxy.config.ssl.ncipher.lib.path STRING	/opt/nfast/toolkits/hwcrhk	Specifies the library path for the nCipher nFast accelerator card. You need only change this variable if you did not use the default path when you installed the card.
proxy.config.ssl.cswift.lib.path STRING	/usr/lib	Specifies the library path for the Rainbow CryptoSwift accelerator card. You need only change this variable if you did not use the default path when you installed the card.
proxy.config.ssl.server_port INT	443	Specifies the port used for SSL communication.

Configuration Variable Data Type	Default Value	Description
proxy.config.ssl.client.certification_level INT	0	Sets the client certification level: 0 = no client certificates are required. Traffic Edge does not verify client certificates during the SSL handshake. Access to Traffic Edge depends on Traffic Edge configuration options (such as access control lists). 1 = client certificates are optional. If a client has a certificate, the certificate is validated. If the client does not have a certificate, the client is still allowed access to Traffic Edge unless access is denied through other Traffic Edge configuration options. 2 = client certificates are required. The client must be authenticated during the SSL handshake. Clients without a certificate are not allowed to access Traffic Edge.
proxy.config.ssl.server.cert.filename STRING	server.pem	Specifies the filename of the Traffic Edge SSL certificate (the server certificate).
proxy.config.ssl.server.cert_chain.filename STRING	NULL	Specifies the file, in a chain of certificates, that is the root certificate recognized by your website.
proxy.config.ssl.server.cert.path STRING	/config	Specifies the location of the Traffic Edge SSL certificate (the server certificate).
proxy.config.ssl.server.private_key.filename STRING	NULL	Specifies the filename of the Traffic Edge private key. Change this variable only if the private key is not located in the Traffic Edge SSL certificate file.
proxy.config.ssl.server.private_key.path STRING	NULL	Specifies the location of the Traffic Edge private key. Change this variable only if the private key is not located in the SSL certificate file.
proxy.config.ssl.CA.cert.filename STRING	NULL	Specifies the filename of the certificate authority that client certificates will be verified against.

Configuration Variable Data Type	Default Value	Description
proxy.config.ssl.CA.cert.path STRING	NULL	Specifies the location of the certificate authority file that client certificates will be verified against.
client related configuration		
proxy.config.ssl.client.verify.server INT	0	Configures Traffic Edge to verify the origin server certificate with the Certificate Authority (CA).
proxy.config.ssl.client.cert.filename STRING	NULL	Specifies the filename of SSL client certificate installed on Traffic Edge.
proxy.config.ssl.client.cert.path STRING	/config	Specifies the location of the SSL client certificate installed on Traffic Edge.
proxy.config.ssl.client.private_key.filename STRING	NULL	Specifies the filename of Traffic Edge private key. Change this variable only if the private key is not located in the Traffic Edge SSL client certificate file.
proxy.config.ssl.client.private_key.path STRING	NULL	Specifies the location of the Traffic Edge private key. Change this variable only if the private key is not located in the SSL client certificate file.
proxy.config.ssl.client.CA.cert.filename STRING	NULL	Specifies the filename of the certificate authority against which the origin server will be verified.
proxy.config.ssl.client.CA.cert.path STRING	NULL	Specifies the location of the certificate authority file against which the origin server will be verified.
ICP Configuration		
proxy.config.icp.enabled INT	0	Sets ICP mode for hierarchical caching: 0 = disables ICP. 1 = allows Traffic Edge to receive ICP queries only. 2 = allows Traffic Edge to send and receive ICP queries. Refer to *ICP Peering, on page 163*.

Configuration Variable Data Type	Default Value	Description
proxy.config.icp.icp_interface STRING	*your_interface*	Specifies the network interface used for ICP traffic. Note: The Traffic Edge installation script detects your network interface and sets this variable appropriately. If your system has multiple network interfaces, check that this variable specifies the correct interface.
proxy.config.icp.icp_port INT	3130	Specifies the UDP port that you want to use for ICP messages.
proxy.config.icp.multicast_enabled INT	0	Enables (1) or disables (0) ICP multicast.
proxy.config.icp.query_timeout INT	2	Specifies the timeout used for ICP queries.
Scheduled Update Configuration		
proxy.config.update.enabled INT	0	Enables (1) or disables (0) the Scheduled Update option.
proxy.config.update.force INT	0	Enables (1) or disables (0) a force immediate update. When enabled, Traffic Edge overrides the scheduling expiration time for all scheduled update entries and initiates updates until this option is disabled.
proxy.config.update.retry_count INT	10	Specifies the number of times Traffic Edge can retry the scheduled update of a URL in the event of failure.
proxy.config.update.retry_interval INT	2	Specifies the delay in seconds between each scheduled update retry for a URL in the event of failure.
proxy.config.update.concurrent_updates INT	100	Specifies the maximum simultaneous update requests allowed at any time. This option prevents the scheduled update process from overburdening the host.
SNMP Configuration		
proxy.config.snmp.master_agent_enabled INT	1	Enables (1) or disables (0) the SNMP agent.

Configuration Variable Data Type	Default Value	Description
proxy.config.snmp.snmp_encap_enabled INT	0	When enabled (1), the SNMP agent runs on the encapsulation port instead of the default port 161. When disabled (0), the SNMP agent can run only on the default port 161. Linux only.
Plug-in Configuration		
proxy.config.plugin.plugin_dir STRING	config/plugins	Specifies the location of Traffic Edge plugins.
WCCP Configuration		
proxy.config.wccp.enabled INT	0	Enables (1) or disables (0) WCCP.
proxy.config.wccp.version INT	1	Specifies the version of WCCP being used: 1 = Version 1.0. 2 = Version 2.0.
WCCP 1.0 variables		
proxy.config.wccp.router_ip STRING	NULL	Specifies the IP address of the router sending traffic to Traffic Edge.
proxy.config.wccp.ethernet_interface STRING	*your_interface*	Specifies the Ethernet interface used to talk to the WCCP 1.0 router. Note: The Traffic Edge installation script detects your Ethernet interface and sets this variable appropriately. If your system has multiple network interfaces, check that this variable specifies the correct interface.
WCCP 2.0 variables		
proxy.config.wccp2.security_enabled INT	0	Enables (1) or disables (0) security so that the router and the Traffic Edge can authenticate each other. (If you enable security in Traffic Edge, you must also enable security on the router. Refer to your Cisco router documentation.)
proxy.config.wccp2.password STRING	NULL	Specifies the password used for authentication. This must be the same password configured on the router. It must be at least seven characters long.

Configuration Variable Data Type	Default Value	Description
proxy.config.wccp2.config_file STRING	wccp_config.xml	Specifies the name of the configuration file, in XML format (located in the Traffic Edge `config` directory), in which you list service groups and router and multicast information.
proxy.config.wccp2.slow_start.enabled INT	0	Enables (1) or disables (0) the slow-start option. Refer to *Slow Start, on page 117*.
proxy.config.wccp2.slow_start.increment INT	10	Specifies the percentage of traffic that the router sends to the leader Traffic Edge node at every interval specified by the variable proxy.config.wccp2.slow_start.interval. Setting this value to 100 percent disables the slow-start option.
proxy.config.wccp2.slow_start.interval INT	15	Specifies the number of seconds after which the leader node increments the percentage of traffic.
proxy.config.wccp2.layer_2_rewrite.enabled INT	0	Enables Layer-2 redirection. Refer to *L2 Redirection, on page 115*. You can set the following values: `0` to disable L2 redirection. `1` to use L2 redirection only when the router advertises L2 for *both* forward mode and return mode; for example, if the router advertises L2 for forward mode but GRE encapsulation for return mode, Traffic Edge does not use L2 redirection. 2 to use L2 redirection even though the router advertises L2 redirection for forward mode but GRE encapsulation for return mode. You cannot configure Traffic Edge to use L2 redirection for forward mode and GRE encapsulation for return mode.

Configuration Variable Data Type	Default Value	Description
proxy.config.wccp2.rev_encapsulation INT	1	Enables (1) or disables (0) reverse encapsulation so that Traffic Edge sends encapsulated returned (bypassed) packets to the router.
ARM (Security Configuration)		
proxy.config.arm.security_enabled INT	0	Enables (1) or disables (0) ARM security. Refer to *Controlling Host Access to the Traffic Edge Machine, on page 204*.

remap.config

The `remap.config` file contains mapping rules that Traffic Edge uses to perform the following actions:

- Map URL requests for a specific origin server to the appropriate location on Traffic Edge when Traffic Edge acts as a reverse proxy for that particular origin server
- Reverse-map server location headers so that when origin servers respond to a request with a location header that redirects the client to another location, the clients do not bypass Traffic Edge
- Redirect HTTP requests permanently or temporarily without Traffic Edge having to contact any origin servers

Refer to *Chapter 7, Reverse Proxy and HTTP Redirects*, for information about redirecting HTTP requests and using reverse proxy.

IMPORTANT After you modify the `remap.config` file, navigate to the Traffic Edge `bin` directory (in Windows, open a Command Prompt window and navigate to the `Traffic Edge bin` directory) and then run the `traffic_line -x` command to apply the changes. When you apply the changes to a node in a cluster, Traffic Edge automatically applies the changes to all other nodes in the cluster.

Format

Each line in the `remap.config` file must contain a mapping rule. Traffic Edge recognizes three space-delimited fields: *`type`*, *`target`*, and *`replacement`*. The following table describes the format of each field.

Field	Description
`type`	Enter one of the following: `map`—translates an incoming request URL to the appropriate origin server URL (HTTP and streaming media reverse proxy). `reverse_map`—translates the URL in origin server redirect responses to point to the Traffic Edge (HTTP and streaming media reverse proxy). `redirect`—redirects HTTP requests permanently without having to contact the origin server. Permanent redirects notify the browser of the URL change (by returning an HTTP status code 301) so that the browser can update bookmarks. `redirect_temporary`—redirects HTTP requests temporarily without having to contact the origin server. Temporary redirects notify the browser of the URL change for the current request only (by returning an HTTP status code 307).
`target`	Enter the origin or *from* URL. You can enter up to four components: *`scheme`*`://`*`host`*`:`*`port`*`/`*`path_prefix`* *`scheme`* can be http, https, ftp, rtsp, or mms.
`replacement`	Enter the destination or *to* URL. You can enter up to four components: *`scheme`*`://`*`host`*`:`*`port`*`/`*`path_prefix`* *`scheme`* can be http, https, ftp, rtsp, or mms.
`tag`	Specifies Real Networks or QuickTime streams. You can enter `RNI` or `QT`.

Examples

The following section shows example mapping rules in the `remap.config` file.

Reverse Proxy Mapping Rules

The following example shows a map rule that does not specify a path prefix in the target or replacement:

```
map http://www.x.com/ http://server.hoster.com/
```

This rule results in the following translations:

Client Request	Translated Request
http://www.x.com/Widgets/index.html	http://server.hoster.com/Widgets/index.html
http://www.x.com/cgi/form/submit.sh?arg=true	http://server.hoster.com/cgi/form/submit.sh?arg=true

The following example shows a map rule with path prefixes specified in the target:

```
map http://www.intranet.y.com/marketing http://marketing.y.com/
map http://intranet.y.com/sales http://sales.y.com/
map http://intranet.y.com/engineering http://engineering.y.com/
map http://intranet.y.com/ http://info.y.com/
```

These rules result in the following translations:

Client Request	Translated Request
http://www.y.com/marketing/projects/manhattan/specs.html	http://marketing.y.com/projects/manhattan/specs.html
http://www.y.com/stuff/marketing/projects/boston/specs.html	http://info.y.com/marketing/projects/boston/specs.html
http://www.y.com/engineering/marketing/requirements.html	http://engineering.y.com/marketing/requirements.html

The following example shows that the order of the rules matters:

```
map http://www.g.com/ http://external.g.com/
map http://www.g.com/stuff http://stuff.g.com
```

These rules result in the following translation.

Client Request	Translated Request
http://www.g.com/stuff/a.gif	http://external.g.com/stuff/a.gif

In the above examples, the second rule is never applied because all URLs that match the second rule also match the first rule. The first rule takes precedence because it appears earlier in the `remap.config` file.

The following example shows a mapping with a path prefix specified in the target and replacement:

```
map http://www.h.com/a/b http://server.h.com/customers/x/y
```

This rule results in the following translation.

Client Request	Translated Request
http://www.h.com/a/b/c/d/doc.html	http://server.h.com/customers/x/y/c/d/doc.html
http://www.h.com/a/index.html	Translation fails

The following example shows reverse-map rules:

```
map http://www.x.com/ http://server.hoster.com/x/
reverse_map http://server.hoster.com/x/ http://www.x.com/
```

These rules result in the following translations.

Client Request	Translated Request
http://www.x.com/Widgets	http://server.hoster.com/x/Widgets

Client Request	Origin server Header	Translated Header
http://www.x.com/Widgets	http://server.hoster.com/x/Widgets/	http://www.x.com/Widgets/

When acting as a reverse proxy for multiple servers, Traffic Edge is unable to route to URLs from older browsers that do not send the `Host:` header. As a solution, set the **Redirect No-Host Header to URL** option in the **Configure/Content Routing/Mapping and Redirection/General** section of Traffic Manager to a page that explains the situation to the user and advises a browser upgrade or provides a link directly to the origin server, bypassing Traffic Edge. Alternatively, you can manually set the variable `proxy.config.header.parse.no_host_url_redirect` in the `records.config` file to the URL to which Traffic Edge will redirect requests with no host headers.

Redirect Mapping Rules

The following rule permanently redirects all HTTP requests for `www.company.com` to `www.company2.com`:

```
redirect http://www.company.com http://www.company2.com
```

The following rule *temporarily* redirects all HTTP requests for `www.company1.com` to `www.company2.com`:

```
redirect_temporary http://www.company1.com http://www.company2.com
```

snmpd.cnf

The `snmpd.cnf` file configures user access to MIB information and trap destinations. It is beyond the scope of this manual to describe all of the SNMP parameters and formats; only the major parameters affecting access control and trap destination are discussed in this section.

Note Traffic Edge supports 64-bit values in SNMP. If you are using an SNMP monitoring tool that does not support SNMPv2c or SNMPv3, you might have trouble viewing all the Traffic Edge values.

IMPORTANT After you modify the `snmpd.cnf` file, you must restart Traffic Edge.

Format

The `snmpd.cnf` file contains a list of configuration parameters. Lines beginning with the `#` symbol are comments. Each configuration parameter is listed along with formatting variables, as in the following example:

```
#Entry type: snmpNotifyEntry
#Format:  snmpNotifyName  (text)
#         snmpNotifyTag  (text)  (keyed on snmpTargetAddr table)
#         snmpNotifyType  (trap(1), inform(2))
#         snmpNotifyStorageType  (nonVolatile, permanent, readOnly)
#snmpNotifyEntry  31 Console trap nonVolatile
#snmpNotifyEntry  32 TrapSink trap nonVolatile
```

Configuring Trap Destinations

You must modify the `snmpd.cnf` file to send traps to each of your monitoring stations.

You must configure the `snmpnotifyEntry` and `snmpTargetAddrEntry` entries for trap destinations. `snmpnotifyEntry` sends traps to a particular host or group of hosts. `snmpTargetAddrEntry` defines the IP addresses for a host or group of hosts.

For example, to send traps to a host named `host_a`, you need an `snmpnotifyEntry` line similar to the following:

```
snmpnotifyEntry 31 host_a trap nonVolatile
```

This line defines a trap destination named `host_a`, which can represent a single IP address or a group of IP addresses. In place of `host_a`, enter the name of the host or group of hosts to receive traps on your system. In place of `31`, enter a unique integer.

Then, for each IP address that you want to define for `host_a`, you must enter a `snmpTargetAddrEntry` line similar to the following. All trap messages destined for `host_a` are sent to the IP addresses defined in the `snmpTargetAddrEntry` lines of the `snmpd.cnf` file.

```
snmpTargetAddrEntry 34 snmpUDPDomain A.B.C.D:0 100 3 host_a v1ExampleParams
nonVolatile 255.255.255.255:0
```

In place of `34`, enter a unique integer. In place of `A.B.C.D`, enter the IP address that you want to define for `host_a`.

Configuring Access Control

By default, read-only access is granted to any host that makes SNMP requests using the community string `public`. To restrict access, you need to remove access-related default entries in the `snmpd.cnf` file and add entries specifying the hosts you want to allow. You must perform the following configuration:

- Define the hosts or host groups for your system (use the `snmpTargetAddrEntry` lines to define the IP addresses associated to each host or host group)
- Define access communities (a community can consist of a host or group of hosts); you need to define hosts before you can define communities
- Give access to the communities that you want to have access; you need to define communities to give them access

Examples

To restrict access, remove the following default `snmpd.cnf` entries, which allow access to any host:

```
vacmAccessEntry snmpv1 public Anyone nonVolatile
vacmAccessEntry snmpv2c public Anyone nonVolatile
snmpCommunityEntry t0000000 public public localSnmpID - nonVolatile
```

To allow access to selected hosts, replace the deleted entries with the following. You can allow access to as many hosts as you want. You can configure one host at a time or one subnet at a time.

For example, suppose you want to allow the single host named `OneHost` to have access to MIB information. You would need the following lines in the `snmpd.cnf` file:

```
snmpTargetAddrEntry 33 snmpUDPDomain A.B.C.D:0 100 3 host_a
v1ExampleParams nonVolatile 255.255.255.255:0
snmpCommunityEntry localSnmpID public public localSnmpID default host_a nonVolatile
vacmAccessEntry OneHost - snmpv1 noAuthNoPriv exact All - All nonVolatile
vacmSecurityToGroupEntry snmpv1 public OneHost nonVolatile
```

The `snmpTargetAddrEntry` line defines the host, `host_a`, which has the IP address `A.B.C.D`. The `communityEntry` line defines the community `OneHost`, which contains the host `host_a`. The `vacmAccessEntry` and `vacmSecurityToGroupEntry` lines allow access to the community `OneHost`.

To allow MIB access to one subnet named `OneNet`, enter the following lines in the configuration file:

Note Use the netmask `255.255.255.0` for the subnet `A.B.C.xxx` in the `snmpTargetAddrEntry` definition.

```
snmpTargetAddrEntry 34 snmpUDPDomain A.B.C.0:0 100 3 net_a v1ExampleParams
nonVolatile 255.255.255.0:0
snmpCommunityEntry localSnmpID public OneNet localSnmpID default net_a nonVolatile
vacmAccessEntry OneNet - snmpv1 no AuthNoPriv exact All - All nonVolatile
vacmSecurityToGroupEntry snmpv1 public OneNet nonVolatile
```

The `snmpTargetAddrEntry` line defines the subnet, `net_a`, which has the IP address `A.B.C.xxx`. The `snmpCommunityEntry` line defines the community `OneNet`, which

contains the subnet `net_a`. The `vacmAccessEntry` and `vacmSecurityToGroupEntry` lines allow access to the community `OneNet`.

socks.config

The `socks.config` file specifies the following information:

- The SOCKS servers through which Traffic Edge must go to access specific origin servers and the order in which Traffic Edge goes through the SOCKS server list

  You can specify your *default* SOCKS servers either in Traffic Manager or by editing the configuration variable `proxy.config.socks.default_servers`. However, the `socks.config` file lets you perform additional SOCKS configuration; you can send requests to specific origin servers through specific SOCKS servers.

- The origin servers you want Traffic Edge to access directly *without* going through the SOCKS server
- The username and password that Traffic Edge uses to connect to a SOCKS server (SOCKS Version 5 only)

IMPORTANT After you modify the `socks.config` file, you must restart Traffic Edge.

Format

To specify the SOCKS servers through which Traffic Edge must go to reach specific origin servers, you must add a rule to the `socks.config` file with the following format:

`dest_ip=`*`ipaddress`* `parent=`*`server_name`*`:`*`port`* `[round_robin=`*`value`*`]`

`ipaddress` is the origin server IP address or range of IP addresses separated by - or `/`.

`server_name` is the hostname of the SOCKS server.

`port` is the port number through which Traffic Edge communicates with the SOCKS server.

`value` is either `strict` if you want Traffic Edge to try the SOCKS servers one by one or `false` if you do not want round-robin selection to occur.

To specify the origin servers you want Traffic Edge to access directly *without* going through the SOCKS server, you must enter a rule in the `socks.config` file in the following format:

`no_socks` *`ipaddress`*

`ipaddress` is a comma-separated list of the IP addresses or IP address ranges associated with the origin servers you want Traffic Edge to access directly.

To specify the username and password Traffic Edge uses for authentication with the SOCKS Version 5 server, you must enter a rule in the `socks.config` file in the following format:

`auth u` *`username password`*

`username` is the username and *`password`* is the password used for authentication.

Note Each rule in the `socks.config` file can consist of a maximum of 400 characters. The order of the rules in the `socks.config` file is not important.

Examples

The following example configures Traffic Edge to send requests to the origin servers associated with the range of IP addresses 123.15.17.1 - 123.14.17.4 through the SOCKS server `socks1` on port 1080 and `socks2` on port 4080. Because the optional specifier `round_robin` is set to `strict`, Traffic Edge sends the first request to `socks1`, the second request to `socks2`, the third request to `socks1`, and so on.

```
dest_ip=123.14.15.1 - 123.14.17.4 parent=socks1:1080;socks2:4080
round_robin=strict
```

The following example configures Traffic Edge to access the origin server associated with the IP address `11.11.11.1` directly *without* going through the SOCKS server:

```
no_socks 11.11.11.1
```

The following example configures Traffic Edge to access the origin servers associated with the range of IP addresses `123.14.15.1 - 123.14.17.4` and the IP address `113.14.18.2` directly *without* going through the SOCKS server:

```
no_socks 123.14.15.1 - 123.14.17.4, 113.14.18.2
```

The following example configures Traffic Edge to use the username `traffic_server` and the password `secret` for authentication with the SOCKS Version 5 server:

```
auth u traffic_server secret
```

splitdns.config

The `splitdns.config` file enables you to specify the DNS server that Traffic Edge should use for resolving hosts under specific conditions.

To specify a DNS server, you must supply the following information in each active line within the file:

- A primary destination specifier in the form of a destination domain, a destination host, or a URL regular expression
- A set of server directives, listing one or more DNS servers with corresponding port numbers

You can also include the following optional information with each DNS server specification:

- A default domain for resolving hosts
- A search list specifying the domain search order when multiple domains are specified

For more information, refer to *Configuring DNS Server Selection (Split DNS), on page 213*.

IMPORTANT After you modify the `splitdns.config` file, navigate to the Traffic Edge `bin` directory (in Windows, open a Command Prompt window and navigate to the `Traffic Edge bin` directory) and then run the `traffic_line -x` command to apply the changes. When you apply the changes to a node in a cluster, Traffic Edge automatically applies the changes to all other nodes in the cluster.

Format

Each line in the `splitdns.config` file uses one of the following formats:

`dest_domain=`*`dest_domain`* | *`dest_host`* | *`url_regex`* `named=`*`dns_server`*
`def_domain=`*`def_domain`* `search_list=`*`search_list`*

The following table describes each field.

Field	Allowed Value
`dest_domain`	A valid domain name. This specifies that the DNS server selection be based on the destination domain. You can prefix the domain with an exclamation mark (!) to indicate the NOT logical operator.
`dest_host`	A valid hostname. This specifies that the DNS server selection be based on the destination host. You can prefix the host with an exclamation mark (!) to indicate the NOT logical operator.
`url_regex`	A valid URL regular expression. This specifies that the DNS server selection be based on a regular expression.
`dns_server`	This is a required directive. It identifies the DNS server for Traffic Edge to use with the given destination specifier. You can specify a port using a colon (:). If you do not specify a port, 53 is used. You can specify multiple DNS servers separated by spaces or by semicolons (;). You must specify the domains using IP addresses in dot notation.
`def_domain`	A valid domain name. This optional directive specifies the default domain name to use for resolving hosts. Only one entry is allowed. If you do not provide the default domain, the system determines its value from `/etc/resolv.conf` in UNIX or from the Registry in Windows.
`search_list`	A list of domains separated by spaces or semicolons (;). This specifies the domain search order. If you do not provide the search list, the system determines the value from `/etc/resolv.conf` in UNIX or from the Registry in Windows.

Examples

Consider the following DNS server selection specifications:

```
dest_domain=internal.company.com named=255.255.255.255:212
255.255.255.254 def_domain=company.com search_list=company.com
company1.com
```

```
dest_domain=!internal.company.com named=255.255.255.253
```

Now consider the following two requests:

```
http://minstar.internal.company.com
```

This request will match the first line and select DNS server `255.255.255.255` on port `212`. All resolver requests will use `company.com` as the default domain, and `company.com` and `company1.com` as the set of domains to search first.

```
http://www.microsoft.com
```

This request will match the second line. Therefore, Traffic Edge selects DNS server `255.255.255.253`. No `def_domain` or `search_list` was supplied, so Traffic Edge retrieves this information from `/etc/resolv.conf` in UNIX or from the Registry in Windows.

ssl_multicert.config

The `ssl_multicert.config` file lets you configure Traffic Edge to use multiple SSL server certificates with the SSL termination option. If you have a Traffic Edge system with more than one IP address assigned to it, you can assign a different SSL certificate to be served when a client requests a particular IP address.

Format

The format of the `ssl_multicert.config` file is:

`dest_ip=`*`ipaddress`* `ssl_cert_name=`*`cert_name`* `ssl_key_name=`*`key_name`*

`ipaddress` is an IP address assigned to Traffic Edge , *`cert_name`* is the filename of the Traffic Edge SSL server certificate, and *`key_name`* is the filename of the Traffic Edge SSL private key.

Note If the private key is located in the certificate file, you do not need to specify the name of the private key.

Examples

The following example configures Traffic Edge to use the SSL certificate `server.pem` for all requests to the IP address 111.11.11.1 and the SSL certificate `server1.pem` for all requests to the IP address 11.1.1.1. The private key *is* included in the certificate files, so no private key name is specified.

```
dest_ip=111.11.11.1  ssl_cert_name=server.pem
dest_ip=11.1.1.1   ssl_cert_name=server1.pem
```

The following example configures Traffic Edge to use the SSL certificate `server.pem` and the private key `serverKey.pem` for all requests to the IP address 111.11.11.1. Traffic Edge uses the SSL certificate `server1.pem` and the private key `serverKey1.pem` for all requests to the IP address 11.1.1.1.

```
dest_ip=111.11.11.1 ssl_cert_name=server.pem ssl_key_name=serverKey.pem
dest_ip=11.1.1.1 ssl_cert_name=server1.pem ssl_key_name=serverKey1.pem
```

storage.config

The `storage.config` file lists all the files, directories, or hard disk partitions that make up the Traffic Edge cache.

IMPORTANT After you modify the `storage.config` file, you must restart Traffic Edge.

Traffic Edge does not use the cache to store streams for Real Networks but uses the filesystem instead; refer to the *Traffic Edge Installation Guide* for more information.

Format

The format of the `storage.config` file is:

`pathname size`

`pathname` is the name of a partition, directory, or file, and *`size`* is the size of the named partition, directory, or file, in bytes. You must specify a size for directories or files. For raw partitions, size specification is optional.

You can use any partition of any size. For best performance, Inktomi recommends the following:

- Use raw disk partitions.
- For each disk, make all partitions the same size.
- For each node, use the same number of partitions on all disks.

Specify pathnames according to your operating system requirements. See the following examples.

IMPORTANT In the `storage.config` file, a formatted or raw disk must be at least 128 MB.

Examples

The following basic example shows 64 MB of cache storage in the `/big_dir` directory:

```
/big_dir 67108864
```

You can use the `.` symbol for the current directory. Here is an example of 64 MB of cache storage in the current directory:

```
. 67108864
```

Solaris Example

The following example is for the Solaris operating system:

```
/devices/sbus@1f,0/SUNW,fas@e,880000/sd@2,0:a,raw
/devices/sbus@1f,0/SUNW,fas@e,880000/sd@2,0:b,raw
```

Note The size is not required, because the partitions are raw.

Linux Example

The following example is for the Linux operating system:

```
/dev/raw_sdb 9105018880
```

Windows Example

The following example is for the Windows operating system:

```
D:\TrafficServer\2.3\cache 67108864
```

trusted-host.config

The `trusted-host.config` file lists destination hosts that are *trusted sources*. Traffic Edge subsequently bypasses virus scanning for objects requested from these destination hosts.

The `trusted-host.config` file is located in the Traffic Edge `config/plugins` directory. The file exists only if you installed the Antivirus Extension.

Note If Traffic Edge does not contain a `trusted-host.config` file or if the file is empty, all hosts are nontrusted sources.

IMPORTANT After you modify the `trusted-host.config` file, you must restart Traffic Edge.

Format

Each line in the `trusted-host.config` file must contain a destination hostname, without the `http://` prefix.

Examples

The following example configures Traffic Edge to bypass virus scanning for objects requested from the origin server `www.myhost.com`:

```
www.myhost.com
```

update.config

The `update.config` file controls how Traffic Edge performs a scheduled update of specific local cache content. The file contains a list of URLs specifying objects that you want to schedule for update.

A scheduled update performs a local HTTP `GET` on the objects at the specific time or interval. You can control the following parameters for each specified object:

- The URL
- URL-specific request headers, which overrides the default
- The update time and interval
- The recursion depth

IMPORTANT After you modify the `update.config` file, navigate to the Traffic Edge `bin` directory (in Windows, open a Command Prompt window and navigate to the `Traffic Edge bin` directory) and then run the `traffic_line -x` command to apply the changes. When you apply the changes to a node in a cluster, Traffic Edge automatically applies the changes to all other nodes in the cluster.

Supported Tag/Attribute Pairs

Scheduled update supports the following tag/attribute pairs when performing recursive URL updates:

- <a href=" ">
- <img src=" ">
- <img href=" ">
- <body background=" ">
- <frame src=" ">
- <iframe src=" ">
- <fig src=" ">
- <overlay src=" ">
- <applet code=" ">
- <script src=" ">
- <embed src=" ">
- <bgsound src=" ">
- <area href=" ">
- <base href=" ">
- <meta content=" ">

Scheduled update is designed to operate on URL sets consisting of hundreds of input URLs (expanded to thousands when recursive URLs are included); it is *not* intended to operate on massively large URL sets, such as those used by Internet crawlers.

Format

Each line in the `update.config` file uses the following format:

`URL``\`*`request_headers`*`\`*`offset_hour`*`\`*`interval`*`\`*`recursion_depth`*`\`

The following table describes each field.

Field	Allowed Inputs
`URL`	HTTP and FTP-based URLs.
`request_headers`	Optional. A list of headers (separated by semicolons) passed in each `GET` request. You can define any request header that conforms to the HTTP specification. The default is no request header.
`offset_hour`	The base hour used to derive the update periods. The range is 00-23 hours.
`interval`	The interval, in seconds, at which updates should occur, starting at Offset hour.
`recursion_depth`	The depth to which referenced URLs are recursively updated, starting at the given URL. This field applies only to HTTP.

Examples

The following example illustrates an HTTP scheduled update:

```
http://www.company.com\User-Agent: noname user agent\13\3600\5\
```

This example specifies the URL and request headers, an offset hour of 13 (1 pm), an interval of one hour, and a recursion depth of 5. This would result in updates at 13:00, 14:00, 15:00, and so on. To schedule for an update to occur only once a day, use an interval value of 24 hours x 60 minutes x 60 seconds = 86400.

The following example illustrates an FTP scheduled update:

```
ftp://anonymous@ftp.company.com/pub/misc/test_file.cc\\18\120\0\
```

This example specifies the FTP request, an offset hour of 18 (6 pm), and an interval of every two minutes. The user must be *anonymous* and the password must be specified by the variable `proxy.config.http.ftp.anonymous_passwd` in the `records.config` file.

vscan.config

The `vscan.config` file specifies the following information:

- The CarrierScan Server to which Traffic Edge sends objects to be scanned
- The amount of time that Traffic Edge keeps a connection open to CarrierScan Server to receive scanning results
- The URL of the custom response page
- The location for temporary files
- The maximum number of objects that Traffic Edge can store in the send queue to CarrierScan Server

The `vscan.config` file is located in the Traffic Edge `config/plugins` directory and exists only if you installed the Antivirus Extension. Traffic Edge must contain the `vscan.config` file for the Antivirus Extension to work.

IMPORTANT After you modify the `vscan.config` file, you must restart Traffic Edge.

Format

Each line in the `vscan.config` file defines a configuration variable with a value. You can change the values to suit your needs.

The configuration variables are described in the following table.

Variable	Description
plugin.nb_threads	Specifies the maximum number of objects that Traffic Edge can send simultaneously to CarrierScan Server. Each thread opens a socket between Traffic Edge and CarrierScan Server. The default value is 20. For information about *scan threads*, refer to the CarrierScan Server documentation.
plugin.temp_path	Specifies where Traffic Edge stores temporary files. The default value is `/tmp`. The directory you specify must exist and must have write permissions. The Antivirus Extension will not operate properly without this variable.
plugin.error_redirection	Specifies the URL of the custom error-response page to which the user is redirected if CarrierScan Server detects an infected file that cannot be repaired. This parameter is optional.
plugin.logging_mask	Specifies the information that Traffic Edge records in the `vscan.log` and `vscan_stats.log` files. You can specify the following values: `T` to record successful scanning transactions in the `vscan.log` file. `I` to record transactions with infected content in the `vscan.log` file. `E` to record transactions with errors in the `vscan.log` file. `S` to record statistics in the `vscan_stats.log` file. `NULL` to disable logging. The following example records all errors and all transactions with infected content in the `vscan.log` file and records statistics in the `vscan_stats.log` file: `plugin.logging_mask=IES` The default value is `TIES` (full logging).
server.address	Specifies the IP address and port of the CarrierScan Server to which Traffic Edge sends objects to be scanned. If you specify multiple servers, you must separate each entry with three semicolons (;;;), as follows: `123.45.6.7:7777;;;123.4.5.6:7777`
server.timeout	Specifies how long (in milliseconds) Traffic Edge keeps each connection to CarrierScan Server open to receive results about the scanned object. The default value is 5000.
server.max_queue	Specifies how many objects Traffic Edge can store in the send queue to CarrierScan Server. The default value is 250.

Examples

The following is an example `vscan.config` file:

```
plugin.nb_threads=20
plugin.temp_path=/tmp
plugin.error_redirection=http://company.com/error_file_infected.html
server.address=Server:0.0.0.0:7777
server.timeout=5000
server.max_queue=250
```

wccp_config.xml

The `wccp_config.xml` file lets you configure WCCP 2.0 options, such as service groups, router information, and multicast mode.

IMPORTANT After you modify the `wccp_config.xml` file, you must restart Traffic Edge.

Format

The following table describes the XML tags nested under the <WccpConfig> tag in the wccp_config.xml file.

Tag	Description
<DefaultInterface>	Specifies the default network interface, which Traffic Edge uses if you do not specify the <Interface> element in the <ServiceGroup> tag. The <DefaultInterface> tag can contain the optional attribute enabled, which can have a value of "yes" or "no".
<ServiceGroup>	Specifies the service group so that the router can send appropriate traffic to Traffic Edge. The <ServiceGroup> tag can contain the attribute enabled, which can have a value of "yes" or "no". The <ServiceGroup> tag contains the following elements: ◆ <Name>, which specifies the name of the service group. This element also has the optional attribute type, for which the value can be either "Dynamic" or "Standard". If you specify "Standard", Traffic Edge ignores most of the other elements, such as <Id> and <Port> (described below). HTTP is the only standard service group specified by the WCCP2 standard. ◆ <Id>, which specifies the service ID for this group. The service ID can be any number between zero (0) and 255. Each dynamic group must have a different number. ◆ <Port>, which specifies the port used for the service group. You can specify a maximum of eight ports (you must specify each port in a separate <Port> element). You do not need to specify the <Port> element if the service group is "standard". ◆ <Protocol>, which specifies if the protocol is TCP or UDP. The default value is TCP. ◆ <Flags>, which specifies the flag for greater control. The default value <Default> is 0x12. You can specify the value as hex or as a decimal number. Refer to the WCCP2 standard for information about the flags allowed. ◆ <Interface>, which specifies an interface for the service group. If you do not specify an interace, Traffic Edge uses the default interface specified in the <DefaultInterface> tag. You can specify multiple interfaces so that the service group is started on each interface (you must specify each interface in a separate <Interface> element).

Tag	Description
<Routers>	Specifies the router IP address. You can specify a maximum of 32 routers. The <Routers> tag can contain the optional attribute enabled, which can have a value of "yes" or "no". The <Routers> tag contains the following element: ◆ <IP>, which specifies the IP address. You can specify multiple IP addresses (you must specify each IP address in a separate <IP> element).
<MulticastAddress>	Specifies the multicast address that Traffic Edge uses to talk to routers. The <MulticastAddress> tag can contain the optional attribute enabled, which can have a value of "yes" or "no". The <MulticastAddress> tag contains the following elements: ◆ <IP>, which specifies the IP multicast address. ◆ <TTL>, which specifies the time to live for multicasting on this address. This element is optional. If you do not specify the time to live, Traffic Edge uses the default value of 1. If both <Routers> and <MulticastAddress> are specified, Traffic Edge uses the multicast address.

Examples

```
<WccpConfig>
<DefaultInterface>
       eth0
</DefaultInterface>

<ServiceGroup>
       <Name>
              HTTP
       </Name>
       <Id> 0 </Id>
       <Port> 80 </Port>
       <Port> 21 </Port>
</ServiceGroup>

<Routers>
       <IP> 123.45.6.1 </IP>
       <IP> 123.45.6.2 </IP>
</Routers>

<Routers enabled="no">
       <IP> 209.131.32.145 </IP>
</Routers>
```

```
<MulticastAddress enabled="no">
       <IP> 224.0.0.100 </IP>
       <TTL> 1 </TTL>
</MulticastAddress>
```

winnt_intr.config

The `winnt_intr.config` file is used for Windows only. It contains a list of network interfaces available on the PC.

Each line in the `winnt_intr.config` file has the following format:

`interface_name ipaddress`

`interface_name` is the name of the network interface. A default name; for example, `intr0` is created automatically during installation by the installation program. If you change the interface name in this file, you must also change any variables in the `records.config` file that contain the interface name: for example, `proxy.config.icp.icp_interface` and `proxy.config.cluster.ethernet_interface`.

`ipaddress` is the static IP address assigned to the network interface.

Specifying URL Regular Expressions (url_regex)

This section describes how to specify a `url_regex`. Entries of type `url_regex` within the configuration files use regular expressions to perform a match.

The following table offers examples to illustrate how to create a valid `url_regex`.

Value	Description
x	Matches the character x.
.	Match any character.
^	Specifies beginning of line.
$	Specifies end of line.
[xyz]	A *character class*. In this case, the pattern matches either x, y, or z.
[abj-oZ]	A *character class* with a range. This pattern matches a, b, any letter from j through o, or Z.
[^A-Z]	A *negated character class*; for example, this pattern matches any character except those in the class.
r*	Zero or more r, where r is any regular expression.
r+	One or more r, where r is any regular expression.
r?	Zero or one r, where r is any regular expression.
r{2,5}	From two to five r, where r is any regular expression.
r{2,}	Two or more r, where r is any regular expression.
r{4}	Exactly 4 r, where r is any regular expression.
"[xyz]\"images"	The literal string [xyz]"images"
\X	If X is a, b, f, n, r, t, or v, then the ANSI-C interpretation of \x; Otherwise, a literal X. This is used to escape operators such as *.
\0	A NULL character.
\123	The character with octal value 123.
\x2a	The character with hexadecimal value 2a.
(r)	Matches an r; where r is any regular expression. You can use parentheses to override precedence.
rs	The regular expression r, followed by the regular expression s.
r|s	Either an r or an s.
#<n>#	Inserts an *end* node causing regular expression matching to stop when reached. The value n is returned.

You can specify `dest_domain=mydomain.com` to match any host in `mydomain.com`. Likewise, you can specify `dest_domain=.` to match any request.

APPENDIX F

Traffic Edge Error Messages

This appendix contains the following sections:

- *Traffic Edge Error Messages*, below describes the warning messages that Traffic Edge sends to the system log file in UNIX or the Event Viewer in Windows.
- *Traffic Edge Alarm Messages, on page 467*, describes the alarm messages that appear in Traffic Manager Monitor pages.
- *HTML Messages Sent to Clients, on page 468*, describes the HTML error messages that Traffic Edge sends to browser clients.
- *Standard HTTP Response Messages, on page 471*, describes the standard HTTP response codes that origin servers send to browser clients.

Traffic Edge Error Messages

The following table lists messages that can appear in system log files (UNIX) or the Event Viewer (Windows). This list is not exhaustive; it describes warning messages that can occur and might require your attention. For information about warning messages not included in the list below, refer to the Support KnowledgeBase on the Inktomi website at `http://support.inktomi.com`.

Traffic Edge Process Fatal

Message	Description
accept port is not between 1 and 65535. Please check configuration	The port specified in the `records.config` file that accepts incoming HTTP requests is not valid.
self loop is detected in parent proxy configuration	The name and port of the parent proxy are the same as that of Traffic Edge. This creates a loop when Traffic Edge attempts to send the request to the parent proxy.

Traffic Edge Warnings

Message	Description
Logfile error: *error_number*	Generic logging error.
Bad cluster major version range *version1-version2* for node *IP address* connect failed	Incompatible software versions causing a problem.
can't open config file *filename* for reading custom formats	Custom logging is enabled, but Traffic Edge cannot find the `logs.config` file.
connect by disallowed client *IP address*, closing	The specified client is not allowed to connect to Traffic Edge. The client IP address is not listed in the `ip_allow.config` file.
Could not rename log *filename* to *rolled filename*	System error when renaming log file during roll.
Did *this_amount* of backup still to do *remaining_amount*	Congestion is approaching.
Different clustering minor versions *version 1*, *version 2* for node *IP address* continuing	Incompatible software versions causing a problem.
log format symbol *symbol_name* not found	Custom log format references a field symbol that does not exist. Refer to *Appendix D, Event Logging Formats*.
missing field for field marker	Error reading a log buffer.
Unable to accept cluster connections on port: *cluster_port_number*	Call technical support.
Unable to open log file *filename*, errno=*error_number*	Cannot open the log file.
Error accessing disk *disk_name*	Traffic Edge might have a cache read problem. You might have to replace the disk.
Too many errors accessing disk *disk_name*: declaring disk bad	Traffic Edge is not using the cache disk because it encountered too many errors. The disk might be corrupt and might have to be replaced.
No cache disks specified in `storage.config` file: cache disabled	The Traffic Edge `storage.config` file does not list any cache disks. Traffic Edge is running in proxy-only mode. You must add the disks you want to use for the cache to the `storage.config` file (refer to *storage.config, on page 455*).

Traffic Edge Alarm Messages

The following table describes alarm messages that you might see in Traffic Manager.

Message	Description
[Rollback::Rollback] Config file is read-only: *filename*	Go to the Traffic Edge `config` directory and check the indicated file permissions; change them if necessary.
[Rollback::Rollback] Unable to read or write config file *filename*	Go to the Traffic Edge `config` directory and make sure the indicated file exists. Check its permissions and change them if necessary.
[Traffic Manager] Configuration File Update Failed: *error_number*	Go to the Traffic Edge `config` directory and check the indicated file permissions; change them if necessary.
[Traffic Manager] Mgmt <==>Proxy conn. closed	This is an informational message informing you that the `traffic_server` process was down.
Access logging suspended - configured space allocation exhausted.	The space allocated to the event log files is full. You must either increase the space or delete some log files to enable access logging to continue. To prevent this from happening, consider rolling log files more frequently and enabling the autodelete feature. Refer to *Rolling Event Log Files, on page 252*.
Access logging suspended - no more space on the logging partition.	The entire partition containing the event logs is full. You must delete or move some log files to enable access logging to continue. To prevent this from happening, consider rolling log files more frequently and enabling the autodelete feature. Refer to *Rolling Event Log Files, on page 252*.
Created zero length place holder for config file *filename*	Go to the Traffic Edge `config` directory and check the indicated file. If it is indeed zero in length, use a backup copy of the configuration file.
Traffic Edge can't open *filename* for reading custom formats	Make sure that the `proxy.config.log2.config_file` variable in the `records.config` file contains the correct path to the custom log configuration file (the default is `logging/logs.config`).
Traffic Edge could not open logfile *filename*	Check permissions for the indicated file and the logging directory.
Traffic Edge failed to parse line *line_number* of the logging config file *filename*	Check your custom log configuration file. There might be syntax errors. Refer to *Inktomi Custom Logging Fields, on page 359*, for correct custom log format fields.
vip_config binary is not setuid root, manager will be unable to enable virtual ip addresses	The `traffic_manager` process is not able to set virtual IP addresses. You must setuid root for the `vip_config` file in the Traffic Edge `bin` directory.

HTML Messages Sent to Clients

Traffic Edge returns detailed error messages to browser clients when there are problems with the HTTP transactions requested by the browser. These Traffic Edge response messages correspond to standard HTTP response codes, but provide more information. A list of the more frequently encountered HTTP response codes is provided on *page 471*. You can customize the Traffic Edge response messages.

The following table lists the Traffic Edge hard-coded HTTP messages, their corresponding HTTP response codes, and their corresponding customizable files.

Title	HTTP Code	Description	Customizable Filename
Access Denied	403	You are not allowed to access the document at location *URL*.	access#denied
Bad HTTP request for FTP Object	400	Bad HTTP request for FTP object.	ftp#bad_request
Cache Read Error	500	Error reading from cache. Please retry request.	cache#read_error
Connection Timed Out	504	Server has not sent any data for too long a time.	timeout#inactivity
Content Length Required	400	Could not process this request because no Content-Length was specified.	request#no_content_length
Cycle Detected	400	Your request is prohibited because it would cause an HTTP proxy cycle.	request#cycle_detected
Forbidden	403	*port_number* is not an allowed port for SSL connections. (You have made a request for a secure SSL connection to a forbidden port number.)	access#ssl_forbidden
FTP Authentication Required	401	You need to specify a correct username and password to access the requested FTP document *URL*.	ftp#auth_required
FTP Connection Failed	502	Could not connect to the server *server_name*.	connect#failed_connect
FTP Error	502	The FTP server *server_name* returned an error. The request for document *URL* failed.	ftp#error

Title	HTTP Code	Description	Customizable Filename
Host Header Required	400	An attempt was made to transparently proxy your request, but this attempt failed because your browser did not send an HTTP `Host` header. Manually configure your browser to use `http://`*`proxy_name`*`:`*`proxy_port`* as an HTTP proxy. Refer to your browser documentation for details. Alternatively, end users can upgrade to a browser that supports the HTTP `Host` header field.	interception#no_host
Host Header Required	400	Your browser did not send a Host HTTP header field and therefore the virtual host being requested could not be determined. To access this website correctly, you will need to upgrade to a browser that supports the HTTP Host header field.	request#no_host
HTTP Version Not Supported	505	The origin server *`server_name`* is using an unsupported version of the HTTP protocol.	response#bad_version
Invalid HTTP Request	400	Could not process this *`client_request`* HTTP method request for *`URL`*.	request#syntax_error
Invalid HTTP Response	502	The host *`server_name`* did not return the document *`URL`* correctly.	response#bad_response
Malformed Server Response	502	The host *`server_name`* did not return the document *`URL`* correctly.	response#bad_response
Malformed Server Response Status	502	The host *`server_name`* did not return the document *`URL`* correctly.	response#bad_response
Maximum Transaction Time exceeded	504	Too much time has passed transmitting document *`URL`*.	timeout#activity
No Response Header From Server	502	The host *`server_name`* did not return the document *`URL`* correctly.	response#bad_response
Not Cached	504	This document was not available in the cache, and you (the client) only accept cached copies.	cache#not_in_cache

Title	HTTP Code	Description	Customizable Filename
Not Found on Accelerator	404	The request for *URL* on host *server_name* was not found. Check the location and try again.	urlrouting#no_mapping
NULL	502	The host *hostname* did not return the document *URL* correctly.	response#bad_response
Proxy Authentication Required	407	Please log in with username and password.	access#proxy_auth_required
Server Hangup	502	The server *hostname* closed the connection before the transaction was completed.	connect#hangup
Temporarily Moved	302	The document you requested, *URL*, has moved to a new location. The new location is *new_URL*.	redirect#moved_temporarily
Transcoding Not Available	406	Unable to provide the document *URL* in the format requested by your browser.	transcoding#unsupported
Tunnel Connection Failed	502	Could not connect to the server *hostname*.	connect#failed_connect
Unknown Error	502	The host *hostname* did not return the document *URL* correctly.	response#bad_response
Unknown Host	500	Unable to locate the server named *hostname*. The server does not have a DNS entry. Perhaps there is a misspelling in the server name or the server no longer exists. Double-check the name and try again.	connect#dns_failed
Unsupported URL Scheme	400	Cannot perform your request for the document *URL* because the protocol scheme is unknown.	request#scheme_unsupported

Standard HTTP Response Messages

The following standard HTTP response messages are provided for your information. For a more complete list, refer to the Hypertext Transfer Protocol — HTTP/1.1 Specification.

Message	Description
200	OK
202	Accepted
204	No Content
206	Partial Content
300	Multiple Choices
301	Moved Permanently
302	Found
303	See Other
304	Not Modified
400	Bad Request
401	Unauthorized; retry
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not acceptable
408	Request Timeout
500	Internal server error
501	Not Implemented
502	Bad Gateway
504	Gateway Timeout

APPENDIX G

FAQs and Troubleshooting Tips

This appendix contains the following sections:

Frequently Asked Questions

The following table lists the frequently asked questions (FAQs) discussed in this section and provides page numbers for your reference.

How do you create a raw disk for the cache if all your disks have mounted file systems?

▼ **To create a raw disk:**

1. As root, enter the following command at the prompt to examine which file systems are mounted on the disk you want to use for the Traffic Edge cache:

   ```
   df -k
   ```

2. In a text editor, open the `/etc/vfstab` file (`fstab` in Linux) and comment out or delete the file system entries for the disk.
3. Save and close the `vfstab` file (`fstab` in Linux).
4. Enter the following command for each file system you want to unmount:

   ```
   umount file_system
   ```

   `file_system` is a file system you want to unmount.
5. Install Traffic Edge. When the installation script prompts you for a cache disk, select the raw disk you just created.

How do disk I/O errors affect the cache and what does Traffic Edge do when a cache disk fails?

If a disk drive fails five successive I/O operations, Traffic Edge considers the drive inaccessible and removes the whole disk from the cache. Normal cache operation continues on all other Traffic Edge disk drives.

How can you tell if Solaris 2.7 or 2.8 is running in 32- or 64-bit mode?

Enter the following command at the prompt to display the supported instruction set architecture (ISA) of the system:

```
isainfo -kv
```

If a client disconnects during the time that Traffic Edge is downloading a large object, is any of the object saved in the cache?

When a client disconnects during an HTTP or FTP operation, Traffic Edge continues to download the object from the origin server for up to 10 seconds. If the transfer from the origin server completes successfully within 10 seconds after the client disconnect, Traffic Edge stores the object in the cache. If the origin server download does not complete successfully within 10 seconds, Traffic Edge disconnects from the origin server and deletes the object from the cache. Traffic Edge does not store partial documents in the cache.

Can Traffic Edge cache Java applets, JavaScript programs, or other application files like VBScript?

Traffic Edge can store and serve Java applets, JavaScript programs, VBScripts, and other executable objects from its cache according to the freshness and cacheability rules for HTTP objects.

Traffic Edge does not execute the applets, scripts, or programs. These objects run only when the client system that sent the request loads them.

How do you access Traffic Manager if you forget the master administrator password?

During installation, you can specify an administrator password. The installer automatically encrypts the password and stores the encryptions in the `records.config` file. Each time you change passwords in Traffic Manager, Traffic Edge updates the `records.config` file.

If you forget the administrator password and cannot access Traffic Manager, you can clear the current password in the `records.config` file (set the value of the configuration variable to `NULL`) and then enter a new password in Traffic Manager. You cannot set passwords in the `records.config` file because the password variables can only contain password encryptions or the value `NULL`.

▼ **To clear and re-enter the administrator password:**

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Set the variable `proxy.config.admin.admin_password` to `NULL` to leave the password blank.

   Ensure that there are no trailing spaces after the word `NULL`.
3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.
6. Log on to Traffic Manager. When prompted for the username and password, enter the administrator ID and leave the password entry blank.

   Because you have already cleared the password in the `records.config` file, you do not need a password to log on as the administrator.
7. On the **Configure** tab, click the **UI Setup** button under **My Proxy**.
8. Click the **Login** tab.
9. In the **Administrator** section, leave the **Old Password** field empty. Type the new password in the **New Password** field and then retype the new password in the **New Password (Retype)** field.
10. Click the **Apply** button.

    The next time you access Traffic Manager, you must use the new password.

How do you apply changes to the logs_xml.config file to all nodes in a cluster?

After you modify the `logs_xml.config` file on one Traffic Edge node, enter the following command from the Traffic Edge `bin` directory:

```
traffic_line -x
```

Traffic Edge applies the changes to all nodes in the cluster. The changes take effect immediately.

In Squid- and Netscape-format log files, what do the cache result codes mean?

The following table describes the cache result codes in the Squid and Netscape log files.

Cache Result Code	Description
TCP_HIT	Indicates that a valid copy of the requested object was in the cache and that Traffic Edge sent the object to the client.
TCP_MISS	Indicates that the requested object was not in the cache and that Traffic Edge retrieved the object from the origin server or from a parent proxy and sent it to the client.
TCP_REFRESH_HIT	Indicates that the object was in the cache but was stale. Traffic Edge made an `if-modified-since` request to the origin server and the origin server sent a `304 not-modified` response. Traffic Edge sent the cached object to the client.
TCP_REF_FAIL_HIT	Indicates that the object was in the cache but was stale. Traffic Edge made an `if-modified-since` request to the origin server but the server did not respond. Traffic Edge sent the cached object to the client.
TCP_REFRESH_MISS	Indicates that the object was in the cache but was stale. Traffic Edge made an `if-modified-since` request to the origin server and the server returned a new object. Traffic Edge served the new object to the client.
TCP_CLIENT_REFRESH	Indicates that the client issued a request with a no-cache header. Traffic Edge obtained the requested object from the origin server and sent a copy to the client. Traffic Edge deletes any previous copy of the object from the cache.
TCP_IMS_HIT	Indicates that the client issued an `if-modified-since` request and the object was in the cache and fresher than the IMS date, or an `if-modified-since` request to the origin server found that the cache object was fresh. Traffic Edge served the cached object to the client.
TCP_IMS_MISS	Indicates that the client issued an `if-modified-since` request and the object was either not in cache or was stale in cache. Traffic Edge sent an `if-modified-since` request to the origin server and received the new object. Traffic Edge sent the updated object to the client.
TCP_SWAPFAIL	Indicates that the object was in the cache but could not be accessed. The client did not receive the object.

Cache Result Code	Description
ERR_CLIENT_ABORT	Indicates that the client disconnected before the complete object was sent.
ERR_CONNECT_FAIL	Indicates that Traffic Edge could not reach the origin server.
ERR_DNS_FAIL	Indicates that the Domain Name Server could not resolve the origin server name, or that no Domain Name Server could be reached.
ERR_INVALID_REQ	Indicates that the client HTTP request was invalid. Traffic Edge forwards requests with unknown methods to the origin server.
ERR_READ_TIMEOUT	Indicates that the origin server did not respond to the Traffic Edge request within the timeout interval.
ERR_PROXY_DENIED	Indicates that client service was denied by access control configuration.
ERR_UNKNOWN	Indicates that the client connected but subsequently disconnected without sending a request.

What does the cqtx field record in a custom log file?

In forward proxy mode, the `cqtx` field records the complete client request in the log file: for example, `GET http://www.company.com HTTP/1.0`. In reverse proxy mode, the `cqtx` field records the hostname or IP address of the origin server because Traffic Edge remaps the request first according to the map rules in the `remap.config` file.

Does Traffic Edge refresh entries in its host database after a certain period of time if they have not been used?

By default, the Traffic Edge host database observes the time-to-live (ttl) values set by name servers. You can reconfigure Traffic Edge to ignore the ttl set by name servers and use a specific Traffic Edge setting. Alternatively, you can configure Traffic Edge to compare the ttl value set by the name server and the ttl value set by Traffic Edge, and use either the lower or the higher value.

▼ **To adjust the host database settings:**

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Set the value of the variable `proxy.config.hostdb.ttl_mode` to:
   - `0` to obey the ttl values set by the name servers.
   - `1` to ignore the ttl values set by name servers and use the value set by the Traffic Edge configuration variable `proxy.config.hostdb.timeout`. Make sure you set this variable to a value appropriate for your needs.
   - `2` to use the lower of the two values (the one set by the name server or the one set by Traffic Edge).
   - `3` to use the higher of the two values (the one set by the name server or the one set by Traffic Edge).

3 Save and close the `records.config` file.

From the Traffic Edge `bin` directory, run the command `traffic_line -x` to apply the configuration changes.

Can you improve the look of your custom response pages by using images, animated gifs, and java applets?

Traffic Edge can only respond to clients with a single text or HTML document. However, you can provide references on your custom response pages to images, animated gifs, java applets, or objects other than text that are located on a web server.

Add links in the `body_factory` template files in the same way you do for any image in an HTML document, with the full URL in the SRC attribute.

Can Traffic Edge run in both forward proxy and reverse proxy mode at the same time?

When you enable reverse proxy mode, Traffic Edge remaps incoming requests according to the map rules in the `remap.config` file. In addition, Traffic Edge serves all requests that do not match a map rule in forward proxy mode. If you want to run in reverse proxy *only* mode, where Traffic Edge does not serve requests that do not match a map rule, you must set the configuration variable `proxy.config.url_remap.remap_required` to 1 in the `records.config` file.

How do you configure Traffic Edge to serve *only* transparent requests?

You can configure Traffic Edge to serve *only* transparent requests and prevent explicit proxy requests from being served in the following ways:

- You can control client access to Traffic Edge from the `ip_allow.config` file by specifying ranges of IP addresses that are allowed to use Traffic Edge as a proxy cache. If Traffic Edge receives a request from an IP address not listed in a range specified in the file, it discards the request. Refer to *ip_allow.config, on page 383*.

- If you do not know the ranges of client IP addresses allowed to access Traffic Edge, you can add rules to the `ipnat.conf` file so that only requests that have been redirected by your Layer 4 switch or WCCP router reach the proxy port. To make a transparent-only Traffic Edge, add rules in the `ipnat.conf` file before the normal redirect service rule to redirect explicit proxy traffic to a port on which no service is listening; for example, if you want Traffic Edge to ignore explicit HTTP requests, add rules above the normal HTTP redirect rule in the `ipnat.conf` file as shown below (where *`ipaddress`* is the IP address of your Traffic Edge system and *`port_number`* is a port number on which no service is listening):

  ```
  rdr hme0 ipaddress port 80 -> ipaddress port port_number tcp
  rdr hme0 ipaddress port 8080 -> ipaddress port port_number tcp
  rdr hme0 0.0.0.0/0 port 80 -> ipaddress port 8080 tcp
  ```

  Add equivalent rules to the `ipnat.conf` file for each protocol service port or separate network interface to be served. After you make changes to the `ipnat.conf` file, you must restart Traffic Edge.

- If your Traffic Edge system has multiple network interfaces or if you configure the Traffic Edge operating system to use virtual IP addresses, you can give Traffic Edge two IP addresses. One address must be the *real* address that Traffic Edge uses to communicate with origin servers and the other a private IP address (for example 10.0.0.1) for WCCP or switch redirection. After you configure the IP addresses, you must add the following variables to the end of the Traffic Edge `records.config` file. Replace *`private_ipaddress`* with the private IP address used for WCCP or switch redirection and *`real_ipaddress`* with the IP address Traffic Edge uses to communicate with origin servers.

  ```
  LOCAL proxy.local.incoming_ip_to_bind STRING private_ipaddress
  LOCAL proxy.local.outgoing_ip_to_bind STRING real_ipaddress
  ```

How does Traffic Edge store multibitrate clips?

An origin server can store multibitrate clips: for example, a movie encoded for streaming at three different bitrates (20 kbps, 100 kbps, and 300 kbps). A client requests only one bitrate at a time. Therefore, Traffic Edge serves and caches only the bitrate in response to that request. If the client connection changes while streaming is in progress: for example, the connection drops from T1 to 56.6 K, Traffic Edge stops serving and caching the initial bitrate and continues at the new bitrate. Traffic Edge supports multibitrate clips automatically; you do not have to perform any special configuration.

Troubleshooting Tips

The following table lists the troubleshooting tips discussed in this section and provides page numbers for your reference.

When you clear the cache, the statistics in Traffic Manager do not reset to zero

The clear cache command (`traffic_server -Cclear`) does *not* reset the statistics to zero in Traffic Manager. To reset the statistics, use the following procedure.

▼ **To reset the statistics to zero:**

1. Access Traffic Manager.
2. Add `clear_stats.html` to the URL in your browser window and press Return: for example, `http://proxy1:8081/clear_stats.html`.

   The Traffic Manager statistics should all show zero.

The throughput statistic is inaccurate in Traffic Manager

Traffic Edge updates the throughput statistic after it has transferred an entire object. For larger files, the byte count increases sharply at the end of a transfer. The complete number of bytes transferred is attributed to the last 10-second interval, although it can take several minutes to transfer the object.

This inaccuracy is more noticeable with a light load. A heavier load yields a more accurate statistic.

You are unable to execute Traffic Line commands

Traffic Line commands do not execute under the following conditions:

- If the `traffic_manager` process is not running

  In UNIX, check if the `traffic_manager` process is running by entering the following command:

  ```
  Solaris: ps -ef | grep traffic_manager
  ```

  ```
  Linux: ps aux | grep traffic_manager
  ```

  If the `traffic_manager` process is not running, enter the following command from the Traffic Edge `bin` directory to start it:

  ```
  ./traffic_manager
  ```

IMPORTANT Inktomi recommends that you always start and stop Traffic Edge with the `start_traffic_server` and `stop_traffic_server` commands to ensure that all the processes start and stop correctly; refer to *Chapter 2, Getting Started*.

- If you are not executing the command from `$TSHome/bin`

  If the Traffic Edge `bin` directory is not in your path, prepend the Traffic Line commands with `./` (for example, `./traffic_line -h`).

- If multiple Traffic Edge installations are present and you are not executing the Traffic Line command from the active Traffic Edge path specified in `/etc/traffic_server`

  Always change to the correct directory by issuing the command:

  ```
  cd `cat /etc/traffic_server`/bin
  ```

You observe inconsistent behavior when one node obtains an object from another node in the cluster

As part of the initial system preparation process, you must synchronize the clocks on all the nodes in your cluster. Minor time differences cause no problems, but differences of more than a few minutes can affect Traffic Edge operation.

Inktomi recommends that you run a clock synchronization daemon such as `xntpd`. You can obtain the latest version of `xntpd` from the following URL:

```
http://www.eecis.udel.edu/~ntp/
```

Web browsers might display an error document with a data missing message

A message similar to the following displays in web browsers:

```
Data Missing

This document resulted from a POST operation and has expired from the
cache. If you wish you can repost the form data to re-create the
document by pressing the reload button.
```

Web browsers maintain their own local cache in memory and/or disk on the client system. Browser messages about documents that have expired from cache refer to the browser local cache, *not* to the Traffic Edge cache. There is no Traffic Edge message or condition that can cause such messages to appear in a web browser.

For information about browser cache options and effects, refer to the browser documentation.

Traffic Edge does not resolve any websites

The browser indicates that it is contacting the host and then times out with the following message:

```
The document contains no data; Try again later, or contact the server's
Administrator....
```

Make sure that the system is configured correctly and that Traffic Edge can read the name resolution file:

- Check if the server can resolve DNS lookups by issuing the `nslookup` command: for example, `nslookup  www.myhost.com`
- Check if the `/etc/resolv.conf` file contains the valid IP address of your DNS server(s).
- On some systems, if the `/etc/resolv.conf` file is unreadable or has no name server entry, the operating system will use localhost as a name server. However, Traffic Edge does not use this convention. If you want to use localhost as a name server, you must add a name server entry for 127.0.0.1 or 0.0.0.0 in the `/etc/resolv.conf` file.
- Check that the Traffic Edge user account has permission to read the `/etc/resolv.conf` file. Change the file permissions to `rw-r--r--` (`644`).

Maximum document size exceeded message in the system log file

The following message appears in the system log file.

```
WARNING: Maximum document size exceeded
```

A requested object was larger than the maximum size allowed in the Traffic Edge cache. Traffic Edge provided proxy service for the oversized object but did not cache it.

You can set the object size limit for the cache by modifying the **Maximum Object Size** field in the Traffic Manager **Configure**/**Subsystems**/**Cache** area or by modifying the

proxy.config.cache.limits.http.max_doc_size variable in the records.config file. If you do not want to limit the size of objects in the cache, set the document size to 0 (zero).

DrainIncomingChannel message in the system log file

The following messages appear in the system log file:

```
Feb 20 23:53:40 louis traffic_manager[4414]: ERROR ==>
[drainIncomingChannel] Unknown message: 'GET http://
www.telechamada.pt/
HTTP/1.0'


Feb 20 23:53:46 louis last message repeated 1 time


Feb 20 23:53:58 louis traffic_manager[4414]: ERROR ==>
[drainIncomingChannel] Unknown message: 'GET http://www.ip.pt/
HTTP/1.0'
```

These error messages indicate that a browser is sending HTTP requests to one of the Traffic Edge cluster ports, either rsport (default port 8088) or mcport (default port 8089). Traffic Edge discards the request. This error does not cause any Traffic Edge problems. The misconfigured browser must be reconfigured to use the correct proxy port.

Traffic Edge clusters work best when configured to use a separate network interface and cluster on a private subnet so that client machines have no access to the cluster ports.

No cop file message in the system log file

The following message appears repeatedly in the system log file:

```
traffic_cop[16056]: encountered "config/internal/no_cop"
file...exiting
```

The file config/internal/no_cop acts as an administrative control that instructs the traffic_cop process to exit immediately without starting traffic_manager or performing any health checks. The no_cop file prevents Traffic Edge from starting automatically when it has been stopped with the stop_traffic_server command. Without such a static control, Traffic Edge would restart automatically upon system reboot. The no_cop control keeps Traffic Edge off until it is explicitly restarted with the start_traffic_server command.

The Traffic Edge installation script also creates a no_cop file so that Traffic Edge does not start automatically. After you have completed installation and configuration, and have rebooted the operating system, use the start_traffic_server command to start Traffic Edge.

Warning in system log file when manually editing vaddrs.config (Linux)

If you manually edit the `vaddrs.config` file on a Linux system as a nonroot user, Traffic Edge issues a warning message in the system log file similar to the following:

```
WARNING: interface is ignored: Operation not permitted
```

You can safely ignore this message. Traffic Edge does apply your configuration edits.

IMPORTANT Inktomi recommends that you always configure virtual IP addresses from Traffic Manager. Manually editing the `vaddrs.config` file can lead to unpredictable results.

Nontransparent requests fail after enabling always_query_destination

The variable `proxy.config.arm.always_query_dest` in the `records.config` file configures Traffic Edge in transparent mode to ignore host headers and always ask for the IP address of the origin server. When you enable this variable, Traffic Edge obtains the origin server IP address from the existing NAT map list rather than trying to resolve the destination hostname with a DNS lookup. As a result, logged URLs contain only IP addresses, not hostnames. However, explicit requests (nontransparent requests, including requests on port 80) fail, as there is no matching map in the NAT list.

The always_query_destination option works only on the primary proxy port.

Traffic Edge is running but no log files are created

Traffic Edge only writes event log files when there is information to record. If Traffic Edge is idle, there might be no log files.

Ensure that you are looking in the correct directory. By default, Traffic Edge creates log files in its `logs` directory. Check the location of the log files in Traffic Manager by examining the **Log Directory** field on the **General** tab in **Configure/ Subsystems/ Logging**. Alternatively, you can check the value of the variable `proxy.config.log2.logfile_dir` in the `records.config` file.

Check that the log directory has read/write permissions for the Traffic Edge user account. If the log directory does not have the correct permissions, the `traffic_server` process is unable to open or create log files.

Check that logging is enabled. In Traffic Manager, examine the **Logging** area on the **General** tab in **Configure/Subsystems/Logging**. Alternatively, you can check the value of the variable `proxy.config.log2.logging_enabled` in the `records.config` file.

Check that a log format is enabled. In Traffic Manager, check that a standard format is enabled on the **Formats** tab of **Configure/Subsystems/Logging** and /or that the custom format is enabled on the **Custom** tab. In the `records.config` file, you select standard formats or the custom format by editing variables in the `Logging Config` section.

Traffic Edge shows an error indicating too many network connections

By default, Traffic Edge supports 8000 network connections: half of this number is allocated for client connections and half for origin server connections. A connection throttle event occurs when client or origin server connections reach 90% of half the configured limit (3600 by default). When a connection throttle event occurs, Traffic Edge continues processing all existing connections but will not accept new client connection requests until the connection count falls below the limit.

Connection throttle events can occur under the following conditions:

- If there is a *connection spike* (if thousands of client requests all reach the Traffic Edge at the same time). Such events are typically transient and require no corrective action.
- If there is a *service overload* (if client requests continuously arrive faster than Traffic Edge can service them). Service overloads often indicate network problems between Traffic Edge and origin servers or indicate that Traffic Edge needs more memory, CPU, cache disks, or other resources to handle the client load.

Examine the MRTG graphs to determine the nature of the connection throttle. In particular, check the Client Connections, TCP Connections, and Client Ops Per Second graphs. You can also check error messages in the system log file, error log file, or event log files.

If necessary, you can reset the maximum number of connections supported by Traffic Edge either by using Traffic Manager (**Networking/Connection Management/Throttling**) or by editing the value of the configuration variable `proxy.config.net.connections_throttle` in the `records.config` file. Do not increase the connection throttle limit unless the system has adequate memory to handle the client connections required. A system with limited RAM might need a throttle limit lower than the default value.

IMPORTANT Do not set this variable below the minimum value of 100.

Low memory symptoms

Under heavy load, the Linux kernel can run out of RAM. The low memory condition can cause slow performance and a variety of system problems. RAM exhaustion can occur even if the system has plenty of free swap space.

Symptoms of extreme memory exhaustion include the following messages in the system log files (`/var/log/messages`):

```
WARNING: errno 105 is ENOBUFS (low on kernel memory), consider a
memory upgrade

kernel: eth0: can't fill rx buffer (force 0)!

kernel: recvmsg bug: copied E01BA916 seq E01BAB22
```

To avoid memory exhaustion, add more RAM to the system or reduce the load on Traffic Edge.

Connection timeouts with the origin server

Certain origin servers take longer than 30 seconds to post HTTP requests, which results in connection timeouts with Traffic Edge. To prevent such connection timeouts, you must change the value of the configuration variable `proxy.config.http.connect_attempts_timeout` in the `records.config` file to 60 seconds or more.

IBM Web servers do not work with Traffic Edge

IBM web servers do not support the TLS (Transport Layer Security) protocol. For IBM web servers to work with Traffic Edge, you must edit a configuration variable. Use the following procedure:

▼ **To configure Traffic Edge to work with IBM web servers:**

1. In a text editor, open the `records.config` file located in the Traffic Edge `config` directory.
2. Edit the following configuration variable:

Variable	Description
proxy.config.ssl.TLSv1	Set this variable to 0 (zero).

3. Save and close the `records.config` file.
4. In UNIX, navigate to the Traffic Edge `bin` directory.

   In Windows, open a command prompt window and navigate to the Traffic Edge `bin` directory.
5. Run the command `traffic_line -x` to apply the configuration changes.

The Inktomi Antivirus Extension does not report a virus

If the Inktomi Antivirus Extension does not report a virus for a requested object, check the `vscan.log` file to ensure that the object was scanned. The `vscan.log` file is located in the Traffic Edge `config` directory.

If there is no corresponding transaction in the `vscan.log` file, check that you have configured Traffic Edge correctly:

- In a text editor, open the `extensions.config` file, which is located in the Traffic Edge `config` directory. Verify that the object's file extension is listed. If the file extension is not listed, add it to the `extensions.config` file.
- In a text editor, open the `trusted-host.config` file, which is located in the Traffic Edge `config` directory. Verify that the file does not include the destination domain or the host for the requested object. Edit the file if necessary.

If there is an entry in the `vscan.log` file for the object that specifies `CLEAN`, CarrierScan Server has not detected the virus. If you believe there is a virus, check that CarrierScan Server is configured correctly. Refer to the CarrierScan Server documentation.

Traffic Edge responds to all scanning requests with a 302 or 503 response code and the message Forbidden (Failed Virus Check)

Traffic Edge sends a 302 or a 503 response code to the client with the message `Forbidden (Failed Virus Check)` under the following conditions:

- When CarrierScan Server detects a virus for a requested object but cannot repair the file.
- When Traffic Edge cannot connect to CarrierScan Server. In this case, Traffic Edge sends a 302 or a 503 response code with the message `Forbidden (Failed Virus Check)` for all requests that require virus scanning. In addition, the `vscan.log` file records entries with the error code `SCSCANSERVER_ERROR`.

Verify that you have configured the variables in the `vscan.config` file correctly. The `vscan.config` file is located in the Traffic Edge `config/plugins` directory. Also, check that CarrierScan Server is running correctly. Refer to the CarrierScan Server documentation for troubleshooting information.

Glossary

alternates

Different versions of the same web object. Some origin servers answer requests to the same URL with a variety of objects. The content of these objects can vary widely, depending on whether a server delivers content for different languages, targets different browsers with different presentation styles, or delivers variable content at different times of the day.

ARM

Adaptive Redirection Module. Used in transparent proxy caching, ARM is a Traffic Edge component that redirects intercepted client traffic destined for an origin server to the Traffic Edge application. Before the traffic is redirected by the ARM, it is intercepted by an *L4 switch* or router.

ASF

Active Streaming Format. A format for multiple streams of synchronized media. ASF is the format for streaming media files with the extensions `.asf`, `.wma`, and `.wmv`.

ASX

The format of the WMT metafile. A text file that contains the URL for an ASF file. The WMT metafile (ASX file) can have the extension `.asx`, `.wax`, or `.wvx`.

cache

Stores copies of frequently accessed objects close to users and serves them to users when requested. See also *object store*.

cache hierarchy

Levels of caches that communicate with each other. All cache hierarchies recognize the concepts of *parent cache* and *child cache*.

cache hit

An object in the cache that can be served directly to the client.

cache miss

An object that is *not* in the cache or that is in the cache but no longer valid. In both cases, Traffic Edge must get the object from the *origin server*.

caching web proxy server

A web proxy server with local cache storage that allows the proxy to fulfill client requests locally, using a cached copy of the origin server's previous response.

CGI

Common Gateway Interface. A set of rules that describe how an origin server and another piece of software (a *CGI program*) located on the same machine communicate.

cgi-bin

The most common directory name on an origin server in which *CGI* programs are stored.

child cache

A cache lower in a *cache hierarchy* for which Traffic Edge is a parent. Child and parent communicate directly. See also *parent cache*.

cluster

A group of Traffic Edge nodes that share configuration information and can act as a single large virtual cache.

Configure mode

One of two modes in *Traffic Manager*. Configure mode lets you configure the Traffic Edge system. See also *Monitor mode*.

Control Protocol

Protocol that a streaming media format uses to set up and control sessions between client and origin server. The control channel is one of two channels needed for media streaming. It relies on an underlying transport protocol, usually TCP. QuickTime and Real Networks use RTSP as their control protocol. WMT uses MMS. See also *Data Transfer Protocol*.

cookie

A piece of information sent by an origin server to a web browser. The browser software saves the information and sends it back to the server whenever the browser makes additional requests from the server. Cookies enable origin servers to keep track of users.

Data Transfer Protocol

Protocol that a streaming media format uses to stream media data. The data channel is one of two channels needed for media streaming. The data transfer protocol relies on an underlying transport protocol, usually UDP. QuickTime, for example, uses RTP as its data transfer protocol. See also *Control Protocol*.

DNS

Domain Name Service. Traffic Edge includes a fast, asynchronous DNS resolver to streamline conversion of hostnames to IP addresses.

explicit proxy caching

A Traffic Edge configuration option, in which client software (typically a browser) must be specifically configured to send web requests to the Traffic Edge proxy.

forward proxy caching

Proxy caching of content requested by web clients. Can be *transparent* or *explicit*.

FTP

File Transfer Protocol. A protocol based on TCP/IP for reliable file transfer.

full clustering

A Traffic Edge cluster distributes its cache across its nodes into a single, virtual object store, rather than replicating the cache node by node. See also *management-only clustering*.

hierarchical live splitting

When a hierarchical deployment for streaming media performs live splitting. See also *hierarchy* and *live splitting*.

hierarchy

Two or more Traffic Edge hosts between the client and the content. One host between the client and the content is a single-level hierarchy.

HTTP

Hypertext Transfer Protocol. The client/server protocol upon which the World Wide Web is based.

ICP

Internet Cache Protocol. A protocol for proxy caches to exchange information about their content.

IP

Internet Protocol. The lowest-layer protocol under TCP/IP responsible for end-to-end forwarding and long packet fragmentation control.

ISP

Internet Service Provider. An organization that provides access to the Internet.

JavaScript

A scripting language commonly used to create web pages. JavaScript is used to make web pages more interactive; for example, to display forms and buttons. JavaScript can be used with Java, but Java is not required for JavaScript to work correctly.

L4 switch

An Ethernet switch that can control network traffic flow using Level 4 rules. The switch can intercept desired client protocol packets and direct them to a proxy for transparent operation.

live passthrough mode
When Traffic Edge passes live streams from the origin server to the client.

live splitting
When Traffic Edge splits one live stream from the origin server into several streams to different clients.

management-only clustering
A Traffic Edge option, in which all nodes in a cluster automatically share configuration information. See also *full clustering*.

MIB
Management Information Base. The set of parameters that an SNMP management station can query in the SNMP agent of a network device (for example, a router). Traffic Edge supports two MIBs: MIB2 (a well-known standard MIB) and the Inktomi proprietary Traffic Edge MIB, which provides more specific node and cluster information.

MMS
Microsoft Media Services. The Windows Media Technologies control protocol. MMS uses either MMST or MMSU as the data transfer protocol. TCP is the underlying transport protocol for the MMST data transfer protocol. UDP is the underlying transport protocol for the MMSU data transfer protocol.

Monitor mode
One of two modes in *Traffic Manager*. Monitor mode lets you view statistics about Traffic Edge performance and web traffic. See also *Configure mode*.

MRTG
Multi Router Traffic Grapher. A graphing tool provided with Traffic Edge that enables you to monitor Traffic Edge performance.

Netscape log format
A standard access log format. Using the Netscape log format, you can analyze Traffic Edge access log files with off-the-shelf log analysis scripts. See also *Squid log format*.

news server
A server that controls access to a news group. A user must connect to a news server to read and post messages to a news group.

NNTP
Network News Transfer Protocol. A protocol used to distribute, inquire, retrieve, and post news articles.

object store
A custom high-speed database, on which Traffic Edge stores all cached objects.

origin server
The web server or media server that contains the original copy of the requested information.

PAC file
Proxy Auto-Configuration file. A specialized JavaScript function definition that a browser calls to determine how requests are handled.

parent cache
A cache higher up in a *cache hierarchy*, to which Traffic Edge can send requests.

plugin
An add-on feature that provides additional functionality to Traffic Edge, such as origin server blacklisting, web content filtering, authentication, and data transformation.

proxy server
See *web proxy server*.

pull splitting
When a host contacts the upstream host to send payload data.

push splitting
When an upstream host sends data to the downstream host with no request from the downstream host.

reverse proxy
A option that allows Traffic Edge to be configured as an origin server for convenient geographical distribution of server content. Reverse proxy also offloads static content service from servers building dynamic content and provides a peak load buffer or *surge protector* for origin servers. Sometimes referred to as *server acceleration*.

router
A device that handles the connection between two or more networks. Routers look at destination addresses of the packets passing through them and decide which route to send them on.

RTSP
Real Time Streaming Protocol. A control protocol used by the Real Networks and QuickTime streaming formats.

server acceleration
See *reverse proxy*.

SNMP
Simple Network Management Protocol. A set of standards used for communication with devices connected to a TCP/IP network. SNMP-compliant devices (agents) store information about themselves in *MIB*s and provide this information to SNMP Managers.

SOCKS
A circuit-level proxy protocol that provides a tunneling mechanism for protocols that cannot be proxied conveniently.

Squid log format
A standard access log format. Using the Squid log format, you can analyze Traffic Edge event log files with off-the-shelf log analysis scripts. See also *Netscape log format*.

SSL
Secure Sockets Layer. A protocol that enables encrypted, authenticated communications across the Internet. Used mostly in communications between origin servers and web browsers.

streaming media
Audio and video that play over the Internet without downloading the complete files that contain them.

streaming media format
A technology encompassing media players, origin servers, and protocols handling streaming media data. Real Networks, Windows Media Technologies, and QuickTime are streaming media formats.

syslog
The UNIX system logging facility.

TCP
Transmission Control Protocol. An Internet Standard transport layer protocol. TCP provides reliable end-to-end communication by using sequenced data sent by IP.

traffic_cop
A Traffic Edge process that periodically monitors the health of the *traffic_server* and *traffic_manager* processes by issuing heartbeat requests to fetch synthetic web pages.

Traffic Line
A Traffic Edge command-line utility that enables you to monitor performance and change configuration settings.

Traffic Manager
The Traffic Edge browser-based interface consisting of a series of web pages that enable you to monitor performance and change configuration settings.

traffic_manager

A Traffic Edge process and the command and control facility. `traffic_manager` is responsible for launching, monitoring, and reconfiguring the *traffic_server* process. It is also responsible for the administration UI, the proxy autoconfiguration port, the statistics interface, cluster administration, and *virtual IP failover*.

traffic_server

A Traffic Edge process that is the cache processing engine of the Traffic Edge product. `traffic_server` is responsible for accepting connections, processing requests, and serving documents from the *cache* or *origin server*.

Traffic Shell

A Traffic Edge command-line tool that enables you to monitor performance and change configuration settings.

transparent proxy caching

A configuration option that enables Traffic Edge to intercept and respond to Internet requests without requiring users to reconfigure their browser settings. It does this by intercepting traffic destined for an origin server and redirecting that traffic through the Traffic Edge cache.

UDP

User Datagram Protocol. The underlying transport protocol by the data transfer protocols.

Underlying transport protocol

A protocol that transports bits. See also *Control Protocol* and *Data Transfer Protocol*

URL

Uniform Resource Locator. The address that defines the route to a file on the web or other Internet facility.

virtual IP failover

An option available to clustered Traffic Edges, in which Traffic Edge maintains a pool of virtual IP addresses that it assigns to the nodes of a cluster. If a node fails, the remaining nodes mask the fault and take over the failed node's virtual interface.

WCCP

Web Cache Control Protocol. A protocol used by Cisco IOS-based routers to redirect traffic during transparent proxy caching.

web proxy server

A proxy server that forwards client requests to *origin server*s. The proxy server can deny requests according to filter rules or security limitations.

web server

A computer that provides World Wide Web services on the Internet. See also *origin server*.

WPAD

Web Proxy Auto-Discovery. A protocol that allows clients to automatically locate a web proxy, providing the benefits of a proxy without the need for explicit client configuration.

Index

D

E

F

N

O

P

Q

R

S

T

U

V

W

X

Portions of Traffic Edge include the following technology:

OpenSSL 0.9.6

The OpenSSL is an open source toolkit licensed under the GNU General Public License. Copyright © 1989, 1991 Free Software Foundation, Inc. 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

Netscape Directory SDK 4.0 for C

Netscape Directory SDK 4.0 for C is available without license fee under the terms of the Netscape ONE SDK End User License Agreement.

Each of the Components is provided on an "AS IS" basis, without warranty of any kind, including without limitation the warranties of merchantability, fitness for a particular purpose and non-infringement. The entire risk as to the quality and performance of the Components is borne by you. Should the Components prove defective or inaccurate, as the case may be, you and not Netscape or its suppliers assume the entire cost of any service and repair. In addition, the security mechanisms, if any, implemented by the Components have inherent limitations, and you must determine that each of the Components sufficiently meets your requirements. This disclaimer of warranty constitutes an essential part of the agreement. SOME JURISDICTIONS DO NOT ALLOW EXCLUSIONS OF AN IMPLIED WARRANTY, SO THIS DISCLAIMER MAY NOT APPLY TO YOU AND YOU MAY HAVE OTHER LEGAL RIGHTS THAT VARY BY JURISDICTION.

Emanate

Emanate® software is used under license agreement with SNMP Research International Incorporated and the relevant third parties.

Tcl 8.3

Tcl software is copyrighted by the Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, and other parties. The following terms apply to all files associated with the software unless explicitly disclaimed in individual files. The authors hereby grant permission to use, copy, modify, distribute, and license this software and its documentation for any purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim in any distributions. No written agreement, license, or royalty fee is required for any of the authorized uses. Modifications to this software may be copyrighted by their authors and need not follow the licensing terms described here, provided that the new terms are clearly indicated on the first page of each file where they apply.

IN NO EVENT SHALL THE AUTHORS OR DISTRIBUTORS BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE, ITS DOCUMENTATION, OR ANY DERIVATIVES THEREOF, EVEN IF THE AUTHORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. THE AUTHORS AND DISTRIBUTORS SPECIFICALLY DISCLAIM ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. THIS SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, AND THE AUTHORS AND DISTRIBUTORS HAVE NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

libdb

LIBD Copyright © 1991, 1993 The Regents of the University of California. All rights reserved. This product includes software developed by the University of California, Berkeley and its contributors.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE AR DISCLAIMED. IN NO EVENT SHALL

THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRET, INCIDENTIAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING LOSS OF USE, DATA, OR PRIFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER ARISING IN CONTRACT, STRICT LIABILITY OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, ENVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

INN

Copyright © 1991, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001 The Internet Software Consortium and Rich Salz. This code is derived from software contributed to the Internet Software Consortium by Rich Salz Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. All advertising materials mentioning features or use of this software must display the following acknowledgement: This product includes software developed by the Internet Software Consortium and its contributors. 4. Neither the name of the Internet Software Consortium nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE INTERNET SOFTWARE CONSORTIUM AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE INTERNET SOFTWARE CONSORTIUM OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MRTG

Multi Router Traffic Grapher (MRTG) is freely available under the terms of the GNU General Public License. Copyright © 1989, 1991 Free Software Foundation, Inc. 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

Libregx

Copyright © 1992, 1993, 1994, 1997 Henry Spencer. All rights reserved. This software is not subject to any license of the American Telephone and Telegraph Company or of the Regents of the University of California.