INTRODUCTION

Welcome to Inktomi's Traffic Server training program. This is your student workbook. In it you will find details about each of the lessons covered by your instructor. Concepts are explored as we discuss each subject and tested as you practice with the hands-on exercises contained in this workbook. At the end of each unit, you will take a spot quiz to make sure you've absorbed the most important subject information. Answers to each quiz are provided in Appendix A.

As you progress through the lessons in your workbook, you may notice a few icons or notes have been placed throughout the text. Here's what these icons mean:

Progress Check summarizes the points you should have learned in the current lesson

Go Slow identifies tasks or subjects that may be a little tricky and deserve extra attention

This training program has been organized to provide you with a general overview of the Traffic Server product first, followed by a more detailed walk-through of the many features and options that are available to you for customizing configuration and monitoring tasks. It concludes with ideas for growth planning, important tips and techniques from the pros at Inktomi and a solutions workshop. Each unit includes a practice lab or quiz to test your understanding:

Unit 1:	**Technology Overview: Design Goals, Innovations & Architecture**
Unit 2:	**Installing the Traffic Server**
Unit 3:	**Configuring the Traffic Server**
Unit 4:	**Monitoring Performance**
Unit 5:	**Maintenance, Performance & Troubleshooting**
Unit 6:	**Using Traffic Line**
Unit 7:	**Solutions Workshop**

What does Inktomi mean?

The company's name, pronounced "INK-to-me," is derived from a Lakota Indian legend about a trickster spider character.

Inktomi is known for his ability to defeat larger adversaries through wit and cunning.

UNIT 1: TECHNOLOGY OVERVIEW

Objectives for this Unit:

- ✓ Understand the role of a caching proxy server
- ✓ Review the key features for Inktomi's Traffic Server
- ✓ Start the Traffic Server
- ✓ Test basic options in configuration mode
- ✓ Test basic options in monitoring mode

What is a Caching Proxy Server?

To understand what a caching proxy server is, it helps to first understand what a standard proxy server is and what a cache is.

A proxy server provides supervised access to the Internet without compromising security. This means that it typically sits on your firewall to handle requests from inside the organization, forwards the request on to the external server and handles the response, passing results back to the internal web client. A proxy server can also be used to provide access to specific data within the firewall. In this way, the proxy server is the liaison between an internal web client and an external web server.

The cache is the amount of memory or disk space set-aside for storing information about documents and images accessed during a session. When you are directly connected from a web client to an external server, and make a document request, a copy of that document and associated images is stored locally on your machine. This copy gets a special temporary name on your file system, in the directory specified for your cache. You can set how much memory and space is allocated for storing cached documents and images. If you want to use the copy again during your session, rather than going out and getting the document from the original source, it is pulled from the cache directory.

It is easy to see the cache at work. The first time you retrieve a page from a server, it will likely take a considerable time to download text and images. As you move through documents, each subsequent new request also takes quite a bit of time. However, when you click the "back" button, you see the previous document you used is loaded immediately. This is because as you continue to access documents, they continue to be added to your cache. This process limits the demand on Internet resources, but it does take up resources on your computer.

The Role of the Caching Proxy Server

A caching proxy server combines both of these concepts into one server. It is a proxy server (acting as a liaison) that includes a very large cache. Instead of documents being "cached" on each individual machine, there is a central cache that can hold hundreds of thousands of documents. As many users in an organization request the same documents and images, the requests are handled directly from the cache to avoid having to seek them again from the original servers. Performance is much better and (in the case of international users who pay by the byte) much less expensive. A caching proxy server is used to improve performance and compensate for expensive or slow Internet links.

While there are clear benefits, they raise some immediate concerns. If you always serve things from the cache, the information may be out of date and if you cache every document that is requested you may fill up your server in no time. Here is where the real benefits of the Inktomi Traffic Server come into play. This is an intelligent server application that takes into account the real challenges of serving to an entire enterprise. The Traffic Server has a robust set of features and options to allow you to tune for your specific needs, ensure timely and complete information is available, and manage the resources that you have at your disposal.

The Traffic Server caches everything, and as it does, it captures information about each document. When a new request is made for the document, it does a very quick check to make sure that the document is still the same. If so, it serves it from the cache; if not it requests a fresh version of anything that has been updated since it was added to the cache. So, if document text has changed, but an image has not, it will serve one part from the cache and the other from the origin server. The Traffic Server allows you to set the rules for keeping information current, including when to assume a document is "stale" and automatically refresh from the origin server. The "expiration/revalidation" model works the same as it would in a standard browser request, by getting page information prior to rendering the page. Additionally, web servers and many document publishers actually assign a "time to live" and the Traffic Server respects these guidelines when determining cache refresh intervals.

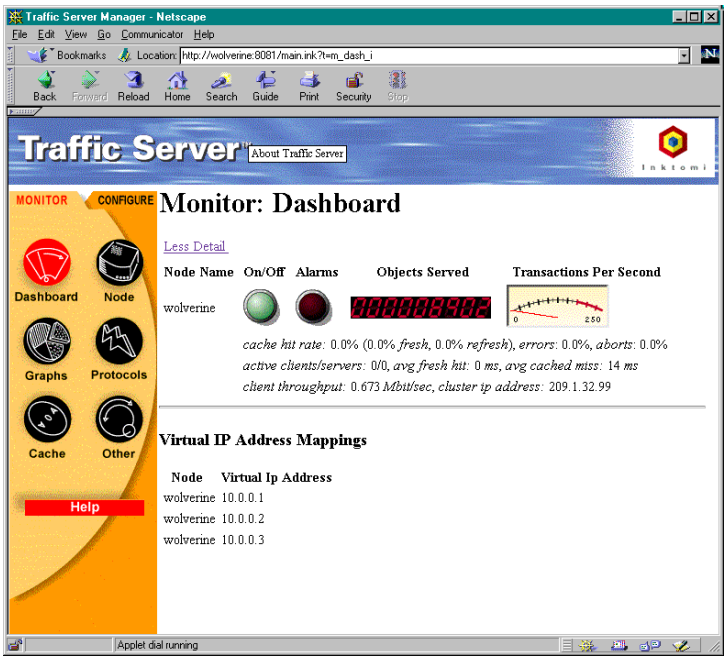
The Traffic Server manages resources. Using a sophisticated technology called *coupled clustering*, you can allow your Traffic Server to expand into a virtual server encompassing as many physical machines (server nodes) as you need to handle the demands of your site. Working together as one unit, the nodes in a virtual server are managed and monitored as one central server.

Thanks to peer-to-peer multicasting, nodes automatically sense overload or failures of other nodes and execute load redistribution or recovery automatically. The cluster distributes the cache across all available nodes without replicating information (as many less efficient caching servers do) to provide an expandable aggregate cache.

The Graphical Administration Manager

The Traffic Server's Graphical Administrator provides secured, single-point administration for a single node or a cluster of nodes. The combinations of administrative tools provided by the Traffic Server Manager allow you to configure, monitor and tune all features and services. Options selected are captured in files that are visible from the command line but safeguarded through the graphical interface. For this reason, Inktomi recommends that you never directly edit configuration files unless you are asked to do so, and guided by Technical Support.

The central point for your administrative duties is called the Dashboard. From here, you can choose to monitor or configure your Traffic Server and you have access to all features and options.



Point your browser at the appropriate port to start The Traffic Server's Administration Manager.

http://myserver:8081
https://myserver:8081

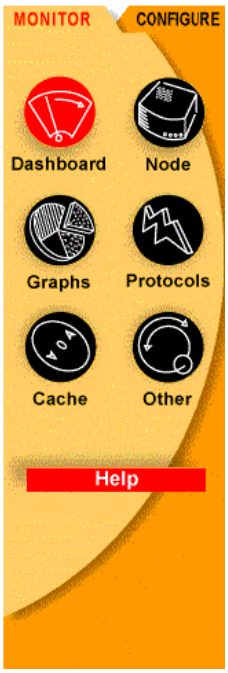
The Traffic Manager reports performance information in a variety of formats, aggregating statistics for the entire cluster, or zooming in on specific activities in a single node.

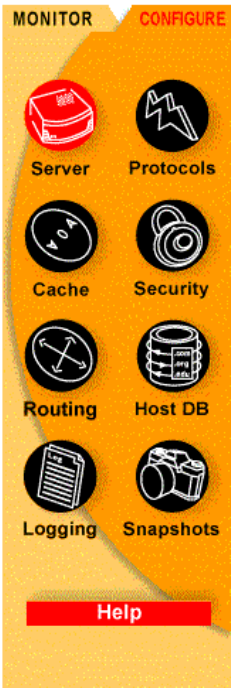
Reviewing system operations is easy. The at-a-glance approach provided by the Dashboard lets you see if all nodes are up and working. If there are problems, alarm lights provide warnings. You can write scripts to determine what is to happen when an alarm goes off.

In addition to administering your local servers, you can perform remote administration secured by encryption. The powerful and centralized logging system allows you to specify the information you wish to track and collates the information on each of the nodes into a single log file. Here is a sample of an entry:

```
Oct 31 02:58:58 eis1 traffic_server[24433]:  NOTE: Logging disk is no longer
low;   access logging resumed.
```

The Traffic Manager organizes key functions in two primary menus: Monitor and Configure. Options appropriately inform you about current status or give you the opportunity to change existing settings for your server. When you install the Traffic Server you have the option of accepting default values for the many Traffic Server settings or to enter your own. These default settings are based on experience and most customers use them initially, tuning as required with real experience at their site.
You can monitor the performance of individual nodes or request server level graphs that depict raw data or performance with respect to time or performance averages.

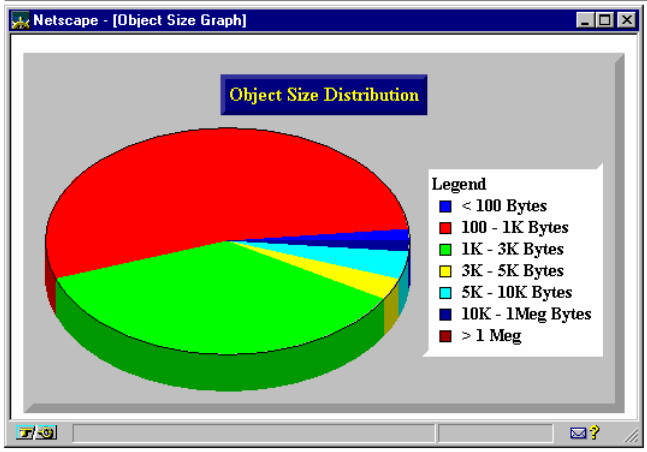




You can review the performance of individual nodes or request graphs that depict either raw data or time and performance averages.

Monitor: Graphs

Graph	Description
Cache Results	Shows the distribution of cache hits and misses
Garbage Collection	Shows distribution of objects collected
Transfer Rates	Shows the speed distribution of transactions
Object Size	Shows the size distribution of Traffic Server responses



The Traffic Server was designed for performance. The engine streams data rapidly and efficiently to and from the disk and network connections. It is able to adapt itself to the particular constraints of your network or disks and can allow many thousands of simultaneous connections.

Because it was originally designed to support multithreading, the engine is able to break large transactions into fast, lightweight processes. These are small, memory efficient tasks (many of which can run without any overhead) so performance remains high even during peak periods. And, thanks to the embedded asynchronous DNS resolver, the conversion of host names to their actual IP addresses is very fast. It does this by actually caching the DNS binding in a distributed host database.

This host database stores information about hosts on the Internet. In addition to the DNS data, it includes information on HTTP version (1.1, 1.0, or 0.9) and the host access frequency to determine which connections to keep alive.

The Object Database

Each node maintains an individual cache of popular objects in a custom, flat-file database. This database includes information about which disks are being used to actually store data objects and an index for locating them. Because the objects are stored in raw disk space, they are seldomly fragmented, even when very large. This method is faster and more efficient because each read and write from the database requires only a single movement of the disk.

Indexes are stored separately from the objects themselves and are cached in memory to reduce index search time. The special "pipeline" architecture allows data to stream in from web hosts to users at the same time as they are being added to the object database.

Scaling Up with Clusters

When you need more cache, you can easily add nodes (additional disks or servers) and "couple" them as a cluster. Once you have created a coupled cluster, the nodes are able to share their contents without replication. You can also establish hierarchical caching which allows you to identify a "parent cache" to speed object retrieval. If a node cannot find the object in its own cluster, it searches the parent cache on another cluster before accessing the Internet to find the object. The parent cache can be any other proxy server.

The Traffic Server automatically stores alternative versions of the same document when they exist (for different languages or browser formats) and intelligently serves the correct version to users based on their browser settings

Maintaining Current Information

You can configure how fresh you want the Traffic Server to keep your documents in its cache, using a variety of criteria in the Object Freshness section of the Cache page. You can tell the Traffic Server to ask the original content server to verify the freshness of objects before serving them when the object has expired, when the object has expired or if the object has no expiration date, always and never. You can specify the minimum freshness information to consider a document cacheable using an explicit lifetime, a "last-modified" lifetime, or nothing. Because some web servers do not stamp their objects with an expiration date, you can set a time (both minimum and maximum) that they can stay in the cache (from

15 minutes to 2 weeks). FTP objects carry no time stamp or date information and will remain in the Traffic Server cache pending removal on a schedule that you specify (from 15 minutes to 2 weeks).

Security

The Traffic Server secures access to the Traffic Manager through authentication. You can turn this on or off and must specify an ID and password when on.

The Traffic Server also supports SOCKS firewall protection and SSL (secure socket layer) encryption. During a proxy connection the SOCKS server grants access based on header information (IP address or port number). You can turn SOCKS on or off, and specify IP or port. Your interface allows you to directly edit the socks.config file.

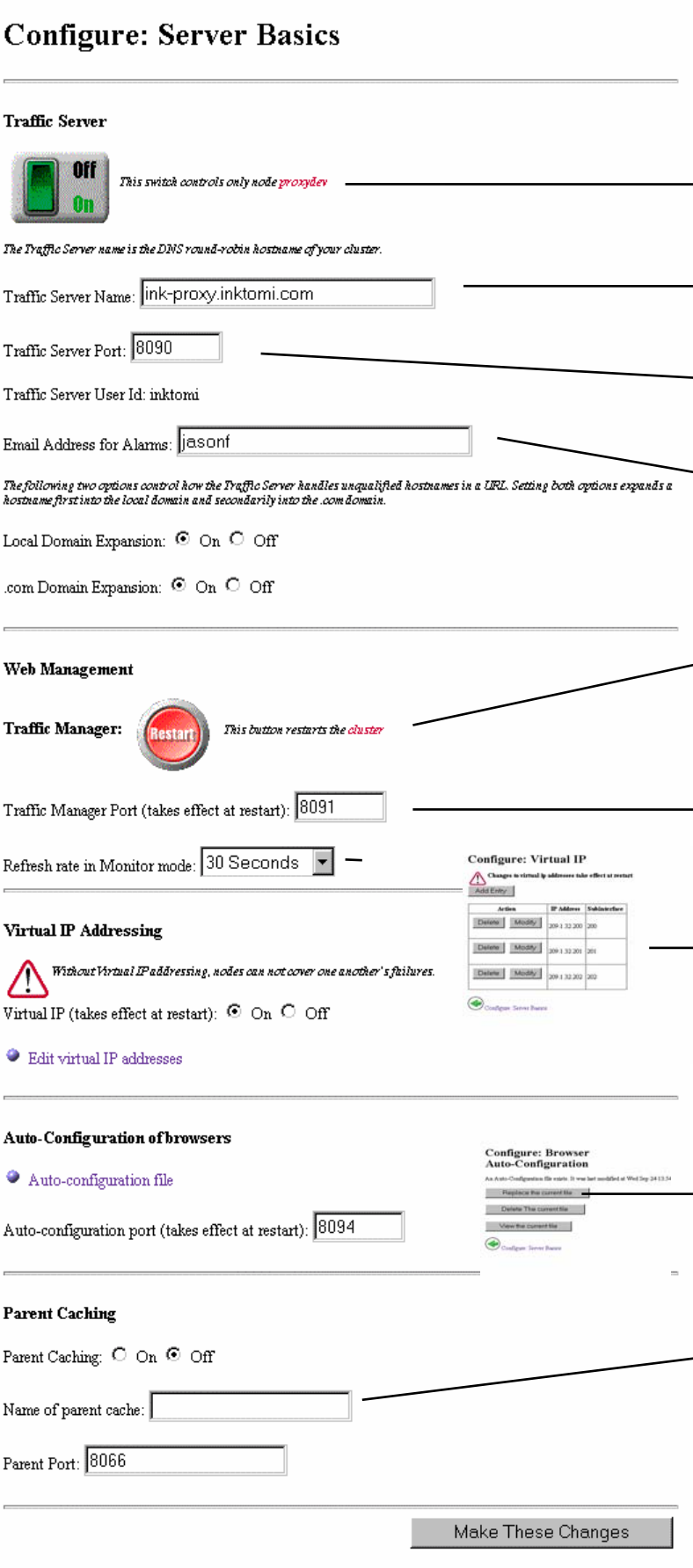
SSL was developed to allow you to send encrypted pictures, text and forms across the multiple networks traveled by data. The client and the web server communicate using SSL through the tunnel provided by the Traffic Server. The Traffic Server does not cache or in any way examine the encrypted data on an SSL connection. Ports that may be used for SSL are configurable on the Security configuration page of Traffic Manager.

Traffic Server also provides for SSL connections to the manager port so that Traffic Manager session can be secure. This feature requires an SSL certificate issued by Inktomi. The certificate location can be configured and SSL can be turned on or off by the Traffic Manager. When SSL is turned "on," Traffic Manager connections must use https:// instead of http:// to connect to the manager port. SSL certificate and configuration for secure Traffic Manager connections is not related to SSL tunneling for secure web connections through the proxy.

Server Configuration Basics

When you install the Traffic Server, some basic configuration values are preset to recommended values. As you learn more about your system, you may want to fine-tune some of these options. It is easy to make changes but it is important to realize that sometimes the change you make may not produce the results you want. There is a great feature called “Snapshots” that allows you to capture the current configuration settings in a file that can be recalled at a moment’s notice. This is like insurance to guard against getting yourself in any serious trouble - but you have to remember to use it. The details of using Snapshots will be covered in Unit 2, when we actually modify values. For now, it is just important to know that the Traffic Server comes with this feature.

Before making any configuration changes, always take a “Snapshot”



Turning off shuts down all caching and proxy services (sometimes required before doing maintenance)

Specify the name for a single node or your entire cluster (used only by Traffic Server)

Specify a dedicated port by which all users can connect to the proxy process. Default is 8080.

Email address to notify when alarms go off

Restarts the Traffic Manager after changes to port numbers or virtual IP addresses (takes 15 secs)

Port number for the Traffic Manager (graphical admin).
Must be dedicated and on the Traffic Server.

When monitoring, how often to refresh screen statistics

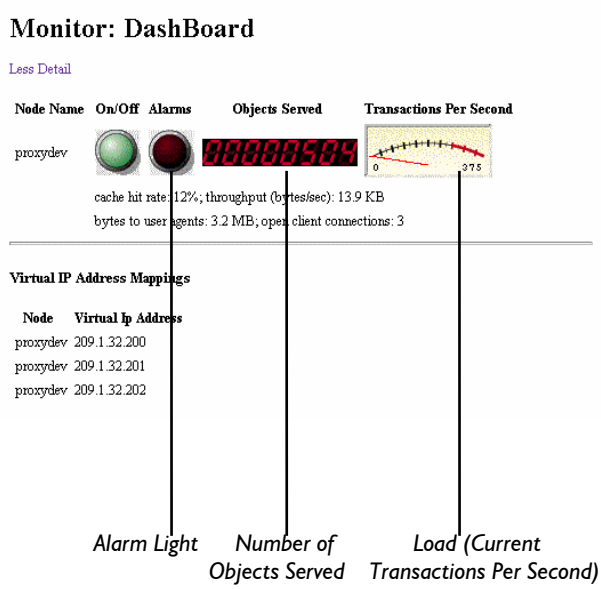
Define virtual IP addresses

Create or edit a script file to automatically configure the user's browser and specify a port to use for downloading the auto-configuration file. (Must also set user's browser options to automatic proxy configuration.)

Enable hierarchical caching by identifying a parent cache(and port) to be searched before going to the Internet

Server Monitor Basics

The Traffic Manager monitors the Traffic Server and reports information about activities and conditions. The first page is called the Dashboard. Here you will see the status of all nodes in your cluster. This uses Java so you need to make sure that your browser is set appropriately.



Clicking the appropriate button accesses the rest of the pages. Be aware that the Dashboard is the only one showing the cluster as a whole, and that the rest of the graphs are oriented to each particular node.

The Dashboard provides information about each node in the cluster and shows alarm status. In addition, you can track the number of objects being served from the cache, cache hit rate, etc.

The Node button provides you with information about the workload on a particular node and compares it with the workload of the overall cluster.

The Graphs button takes you to a list of graphical reports you can select for information about the contents and performance of your server.

The Protocols button shows activities for HTTP, FTP, NNTP and ICP connections.

The Cache button shows activities in a node's cache space and provides information about the cache contents.

The Other button provides access to information or activities related to the host database, DNS performance, cluster performance, SOCKS connections, and logging.

As you can see, getting started with the Inktomi Traffic Server is easy. You install the software, responding to prompts about your environment, and the Traffic Server configures your server with recommended default values.

You can instantly begin monitoring and as you learn more about your particular system, you can add new features or tune existing ones. In the units that follow, you will have the opportunity to explore in detail the installation options and various tuning and customizing parameters for managing your Traffic Server.

Progress Check

By now you should be able to:

1. ***Recognize differences between a caching proxy server and a standard proxy server***
2. ***Describe how caching is handled in Inktomi's Caching Proxy Server and identify key features***
3. ***Log onto the Traffic Manager***
4. ***Switch between configuration and monitor modes***
5. ***Recognize the main purpose for each of the key components available.***

Unit 1 Practice Lab

Objectives for Unit 1 are to understand the basic role of a caching proxy server and to learn about the many key features of Inktomi's Traffic Server and to become familiar with the working environment. In this Practice Lab you will have the opportunity to:

- Start the Traffic Manager
- Review the basic configuration set up during a standard installation
- Practice with basic monitoring of the Traffic Manager

1. Notice the tag on the lower right corner of your monitor. It contains information about your server and user account. Log in to your student account (no password is required) and open a terminal window (right mouse, tools, terminal). This will place you at the top of your Inktomi student directory **`(/space/inktomi/class/train`*`n`*`.)`** Move up one directory (**`cd ..`**) to see the student environment. . Type "**`ls`**" to see the subdirectories. Notice that there are many student directories here and a teacher directory. Your instructor will use the **teacher directory**. When you are more experienced with the product, we will introduce a few problems for you to troubleshoot – so no fair peeking!

2. Return to your student directory and always be sure this is where you are working. The **class_software directory** is a link to the current release of the Traffic Server software. You will use the install script from this directory to actually install the product in upcoming labs. You will be installing and uninstalling the product several times in the course of this class to let you gain experience with different Traffic Server configurations. Your student directory contains a special **class_tools directory** with programs for use during your training. These include a synthetic client and server (which will allow us to populate the cache as if users were actually using the Traffic

Server. There are a few scripts to initiate these programs. There is also a script that deletes a Traffic Server installation, removing all unnecessary files (deleteTS.sh).

Move up two directories to the main inktomi directory. Here you will see the current installation directory for Traffic Server 2.1. We have installed the product for your first lab and you will re-install it in subsequent labs. Move now to the 2.1 directory and `'ls'` to show these files and directories. The **bin directory** contains the executables to run the Traffic Server and the programs to start and stop the Traffic Server processes. The **config directory** contains the actual configuration files which store your specific parameters for your Traffic Server application. The **logs directory** includes an error log (error.log), an alarm log (traffic.out) and the access log (squid.log). The **ui directory** contains the html pages and images used by the Traffic Server. Return to your student directory.

3. Start Netscape and enter the URL for the Traffic Manager: **`http://{your IP address}:8081`**. You can also substitute your server name (for example train1-east, train1-west or train1-remote). You will be prompted for a username and password (use admin for both).

4. When the Traffic Manager starts up, edit your browser preferences to select the Traffic Manager as your "home" page (Edit → Preferences → Navigator and click on "Use Current Page") or create a bookmark for your admin page. The home page is called the Dashboard. Take a few minutes to review the information presented on the Dashboard. Select the "More Detail" option and review the information presented. We are using a special tool that simulates user activity so you can get an idea about the feedback presented by Traffic Server.

   Enter the node name: _____. In a cluster you would see all nodes here.

5. How many objects have been served since the server was last started? _____

6. Click on "Node" to learn more about the current levels of activity.

7. Try each of the monitor buttons in the left panel. We will review the details behind the various options in the coming sections, but for now it is good to move through each and notice what type of information is organized under the various pages. Keep in mind that there is only a synthetic server running, so there won't be a lot of traffic to observe yet.

8. Return to the Dashboard and select the Configure menu. The first page presents Server Basics. Using the information in this unit, go through each of the options listed on this page, to understand what they refer to. When you have finished reviewing all of the information on the Server Basics page, use the buttons in the left panel to again review the information contained in these key component areas like Protocols, Cache, etc. Do not make changes yet, as we will practice with individual settings in another exercise. For now it is just important to gain familiarity with your options and how these options are organized for your use.

Spot Quiz

For each question, circle the letter of the correct answer or answers. Some questions may have more than one right answer.

1. When you request a document from the Internet that is cached on your local machine, what actually happens?

    a. A pointer to the document is saved as a bookmark
    b. A copy of the document is saved on your file system using a special temporary name
    c. Your browser uses the copy of the original document when asked to view it again
    d. Limits demand on Internet resources

2. When you request a document from the Internet that is cached on the Traffic Server, what actually happens?

    a. A report is created listing every document you accessed.
    b. A copy of the requested document is added to a special database that can serve the object locally, when it is deemed current according to Traffic Server guidelines.
    c. A pointer is made to all the individual users' caches so a document can be found somewhere in the organization
    d. All authorized users can share objects cached, directly from the Traffic Server's cache.

3. How does the Traffic Server keep information stored in the cache current?

    a. It tracks modification dates and cache dates in the database and does a quick check to make sure the document can be served locally and still be current.
    b. It throws everything away at midnight and fetches all the documents again when no one is looking
    c. It uses the settings you specify to determine what is current and then checks documents on the origin server to see if there is a more current version?

4. What happens when one Traffic Server node in a cluster goes down?

    a. A report is created that lists all the documents that couldn't be cached so you can cache them later
    b. An alarm goes on and caching stops for that node.
    c. The other nodes in the cluster notice the problem and automatically pick up the extra work.
    d. A node can never go down.

INSTALLING INKTOMI'S TRAFFIC SERVER

- ✓ Review Installation Alternatives
- ✓ Learn About Pre-Installation Requirements
- ✓ Install the Traffic Server
- ✓ Review the Application Environment
- ✓ Learn How Traffic Server is Configured

Installation Alternatives

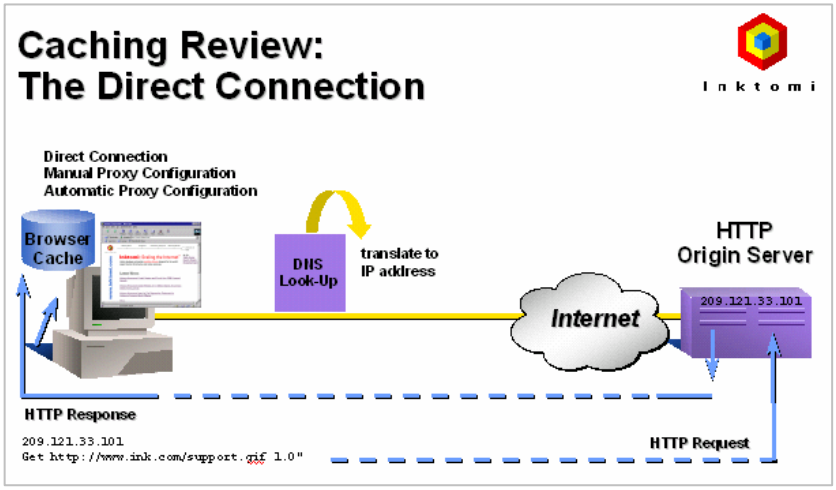
The Inktomi Traffic Server can be installed in a number of specialized configurations. Each configuration determines how the Traffic Server will perform and installation options create the appropriate environment. This caching review will help you understand the role of the Traffic Server in these various configurations.

Caching Review

A Direct Connection

This direct connection shows basic caching without Traffic Server. A request is sent from the browser, the DNS lookup occurs, and the request is directed to the appropriate origin server. The request is fulfilled, returned to the requesting user and the various components in the request (text and images) are cached on the user's machine.

In this example, the user's browser options would be set to a direct connection.



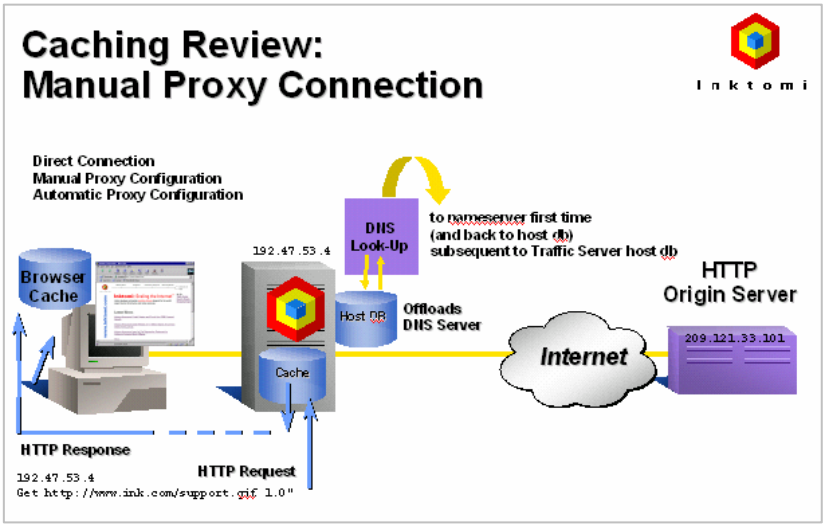
Manual Proxy Connection

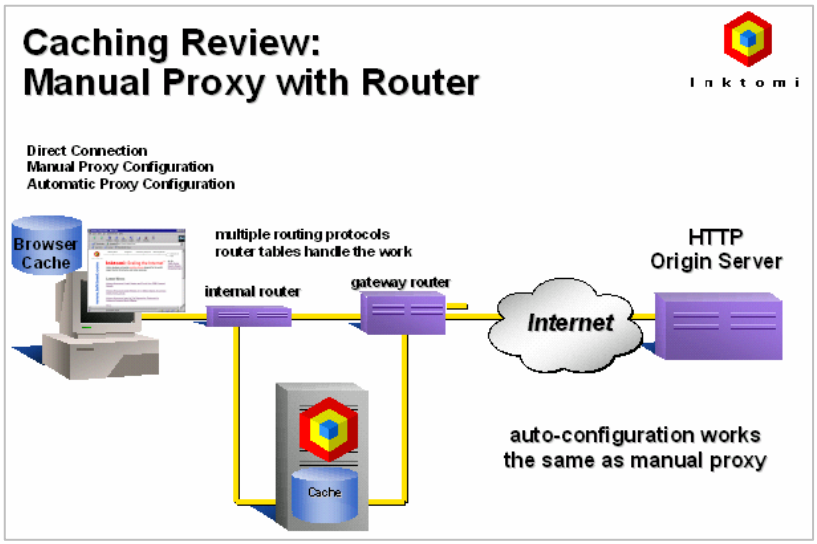
This manual connection shows caching with the Traffic Server.

A request is sent from the browser through the Traffic Server's proxy port. On the first request, The Traffic Server does the DNS lookup (and caches the name resolution in the Traffic Server's Host Database) then directs the request to the appropriate origin server.

The request is fulfilled, being added to the Traffic Server cache at the same time it is being returned to the requesting user. The various components in the request (text and images) are cached on the Traffic Server machine and all subsequent requests are served from this cache until "freshness rules" expire. When this occurs, the old entry will be deleted and a new copy will be fetched.

In this example, the user's browser options would be set to manual proxy connection to a specific port.





Transparency (Solaris Only)

Setting up a Traffic Server using the "transparency" options, allows you to route requests through a layer 4 switch to the Traffic Server. Independent of settings in the user's browser, this method ensures all traffic begins with the Traffic Server.

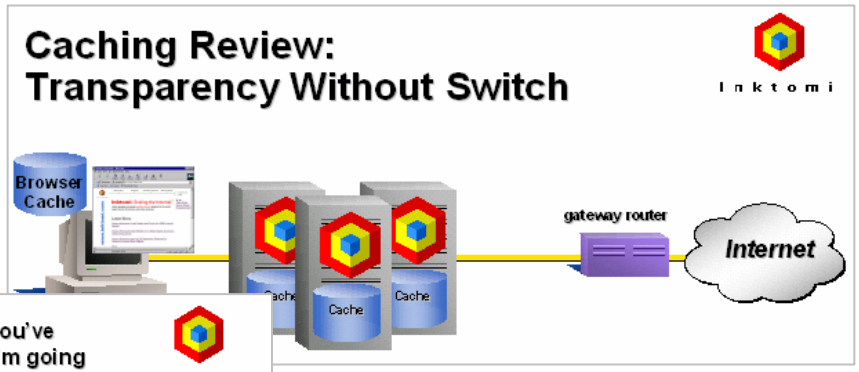
The layer 4 switch provides many routing features and options to help you manage how requests will be processed.

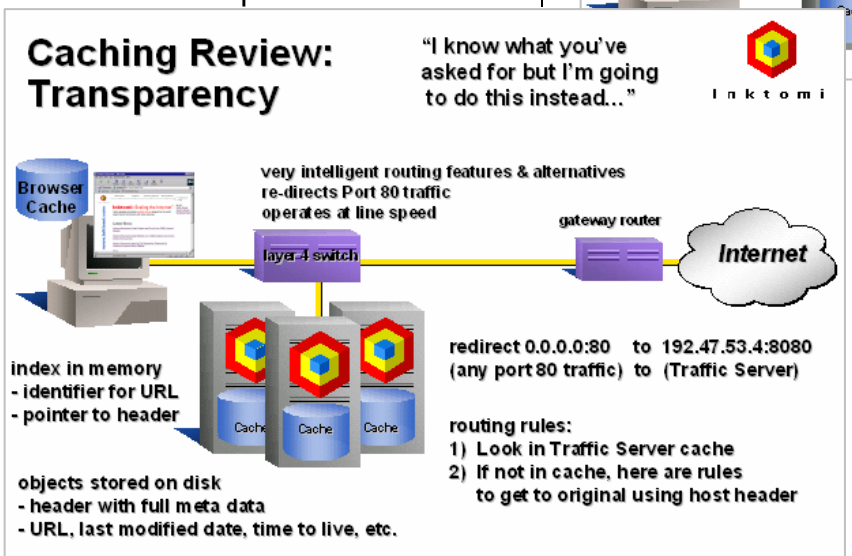
This picture also shows the use of a cluster. A cluster is a groups of servers who share the workload. A key feature in Inktomi's technology is the ability to grow your Traffic Server cache as your application needs grow.

The index of objects stored in the cache is held in memory and the objects themselves are written to your raw disk partitions.

Routing rules dictate how requests are to be managed.

You can install transparency without a layer 4 switch. Redirection is handled through software. Be aware that this loads lots of work on Traffic Server to do router work and that additional software is required to handle port redirection. You must write policy-based rules and require browsers that do host/header. This is slower opening the gateway and has a greater potential for problems.





Reverse Proxy (Web Server Acceleration)

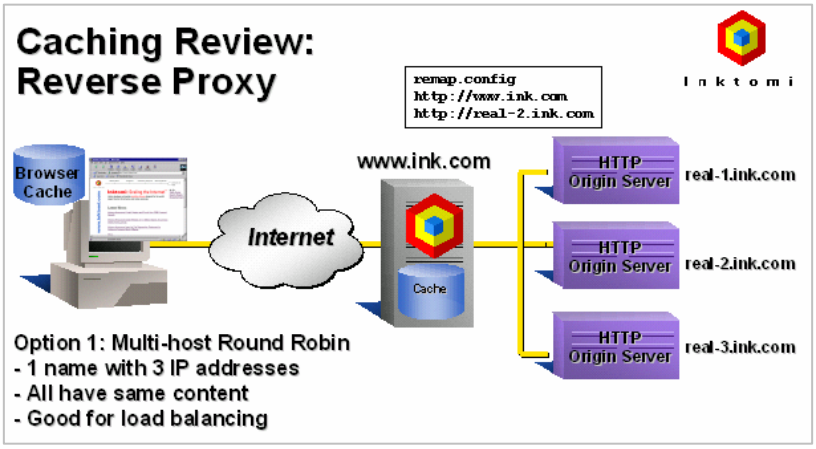
Normally, Traffic Server handles arbitrary web requests to distant web servers, on behalf of a set of users.

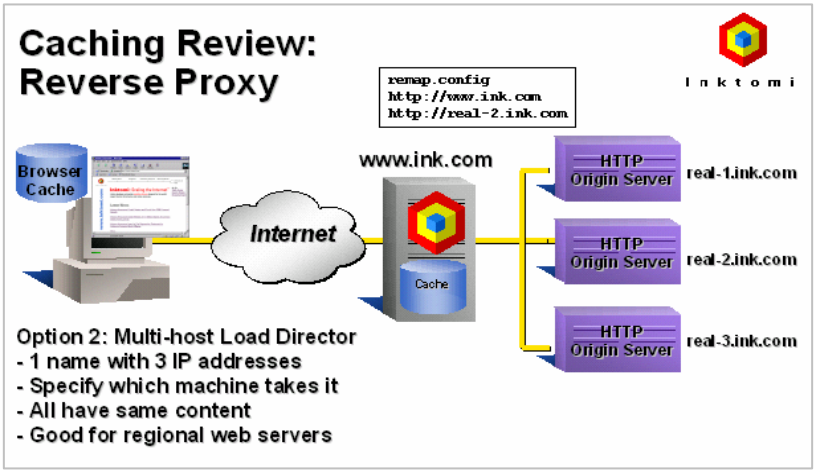
Server acceleration (also known as reverse proxy caching or virtual web hosting) is slightly different.

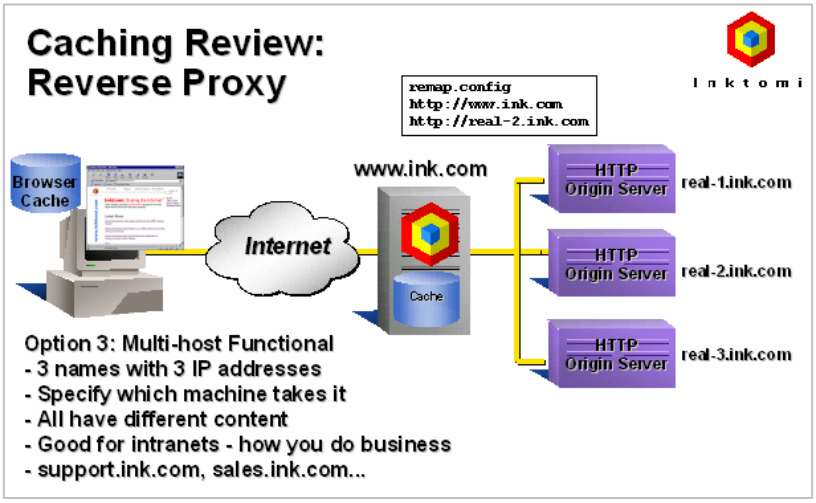
In server acceleration, the Traffic Server IS the web server that others are trying to connect to. The web server hostname resolves to the Traffic Server -- impersonating the actual origin server.

If Traffic Server has the desired object in cache it serves it immediately. If not,it requests the document from the backup server that has the content. A configuration table specifies which backup web server to use to find this content.

Having Traffic Server absorb the main web server request traffic can improve the speed and quality of web serving, reducing load on the backup web servers, while maintaining the publishing environment on the backup web servers.







Preparing to Install Traffic Server

Before you can install the Traffic Server, you need to prepare your target installation node (installation is the second of a two step process). If you are going to use clustering, you will prepare each of the servers that will be a part of your Traffic Server cluster. Preparing the target installation consists of these steps:

- Creating a non-privileged user account (by default this is **`inktomi`** and it will be created automatically during install if you have not already created this account.) When you create the account beforehand, you can set up group rights, account name, etc. This account is used for the Traffic Server daemon and the traffic_manager and traffic_cop processes.

- Verifying your host system meets minimum requirements and that your available disk has no mounted file system and no swap partition. The entire raw disk will be used for the Traffic Server cache. (sized from 2 GB to 16 GB). The disk will be formatted using the default backup partition. You should not need to repartition your disks unless you have already reformatted your disk. If so, reformat first, using the default parameters'

- Labeling the disks for the Traffic Server cache.

- Enabling DNS. Configure DNS by adding at least one nameserver entry in /etc/resolv.conf (for the Traffic Server)

Installing the Traffic Server Software

You must be root to install the Traffic Server because the installation program will create setuid programs and system file modifications. If running transparently, drivers will be installed. When you run the installation program, the appropriate files are chosen for your operating system.

You will need at least 100 MB of free disk space for the installation and another 100 MB of free space for logging (though 500 MB is recommended). The installer will prompt you for target directories and a number of configuration settings including:

- Disk location for logs
- Port mappings
- Email address, username & password for administrator
- Take advantage of network interfaces (cluster only)
- Set a multicast group address (cluster only) for simultaneous transmissions to Traffic Server nodes
- Decide if you will set up Traffic Server to run transparently

- Decide if you will set up Traffic Server as a web server accelerator (reverse proxy)
- Configure Traffic Server Cache

You install the Traffic Server software to function on as a cluster by conducting a separate install for each node of the cluster and then configuring all of the nodes to use identical ports. When you are ready to bring the cluster into service, you start the traffic Server process on each node.

Here is an example of a standard installation:

```
ts-sgi2:traffic_210-5>./install.sh

#############################################################################
#
#        Traffic Server 2.1 Installation
#        This script installs the Traffic Server
#        on system ts-sgi2.
#
#############################################################################

Traffic Server environment configuration
----------------------------------------
Enter an account name for the Traffic Server: [inktomi]

Using account inktomi for Traffic Server install.

Enter the full path of the directory in which to install Traffic Server.
>/usr/people/inktomi
Enter the full path of the directory in which to store
Traffic Server log files: [/usr/people/inktomi/logs] >
Traffic Server network configuration
-----------------------------------
Is this installation part of a Traffic Server cluster [y/n]? n


Would like to configure Traffic Server as a server accelerator (reverse proxy)
[y/n]?n

Traffic Server makes use of 10 ports on your server.
Please enter the starting port number: [8080]
default starting port number 8080 will be used

You have made the following port selections:
-------------------------------------------
1.  Traffic Server Proxy Port   8080
2.  Web  Administration port    8081
3.  Dynamic graphing port       8082
4.  Auto config port            8083
5.  Process manager port        8084
6.  Logging server port         8085
7.  Clustering port             8086
8.  Secondary clustering port   8087
9.  Reliable service port       8088
10. Multicast port              8089
11. Real Networks Client port   8090

***Verifying port assignment conflicts***
The port assignment check has found no conflicts

Enter the port assignment you would like to change (1-10)
```

```
'0' for no changes, 'h' for help
> 0

Traffic Server administration information:
------------------------------------------
Enter an e-mail address for Traffic Server alarm notification: [inktomi]
> Using notification email address inktomi

Enter the Traffic Server administrator user name. This name is not a Unix user
account name, and is only for the Traffic Manager web-based administration program:
[admin]
>
Traffic Manager administrator name admin
Enter the Traffic Server administrator password: >
Enter the Traffic Server administrator password again >

Traffic Server cache configuration:
-----------------------------------
Checking available space for cache.  You cannot use any disks that include a mounted
file system or swap partition for cache storage. Only disk drives not used for any
other purpose are listed for cache selection. Partition 7 of the 'option' disk
normally spans the entire disk drive, and will be used to identify drives for cache
storage. Ready to configure the Traffic Server cache. Select available disk resources
to use for the cache. Remember that space used for the cache cannot be used for any
other purpose.

Here is the list of available disk drives :
(1) /dev/rdsk/dks0d2s7

Choose one of the following options:
(a)   ADD a cache storage location.
(r)   REMOVE a cache storage location.
(s)   SELECT ALL cache storage locations.
(d)   DONE with selection, continue Traffic Server installation.
(q)   QUIT from Traffic Server installation now.

OPTION: s

Here is an updated list of your choice of disk drives:

[X] (1) /dev/rdsk/dks0d2s7

Choose one of the following options:
(a)   ADD a cache storage location.
(r)   REMOVE a cache storage location.
(s)   SELECT ALL cache storage locations.
(d)   DONE with selection, continue Traffic Server installation.
(q)   QUIT from Traffic Server installation now.

OPTION: d
You are quitting from disk drive selection.

Here is the final choices of disk drives for your cache storage configuration:
/dev/rdsk/dks0d2s7

Configuration for cache storage is done.
Installing Traffic Server 2.1 files to /usr/people/inktomi
tar: blocksize = 16
```

```
tar: blocksize = 16
lock - No such file or directory

/usr/people/inktomi/config/snmp-start.config - No such file or directory
./install.sh[2048]: /tmp/TSInstall/snmp-start.config: cannot open: No such file or
directory
/tmp/TSInstall/snmp-start.config.new - No such file or directory
/usr/people/inktomi/config/snmp-start.config - No such file or directory
Cannot access /usr/people/inktomi/config/snmp-start.config: No such file or directory
/dev/rdsk/dks0d2s7 cache partition
Configuring Traffic Server cache. This may take a few minutes.
Do not interrupt cache configuration or you will have an unusable cache.
CLEAR

Clearing Configuration
Clearing Host Database
Clearing Cache

CLEAR, succeeded
Your Traffic Server 2.1 installation is complete.
Please reboot this system before starting Traffic Server.

To start Traffic Server, login as inktomi and enter the command
        start_traffic_server

A log file of this installation process has been written to
/usr/people/inktomi/TSinstall.log

Please consult the Traffic Server User's Guide for full operating information.
ts-sgi2:traffic_210-5>
```

Application Environment – After Installation

The directory structure created by the Traffic Server installation is shown below. A more complete description of key areas follows this diagram:

```
+---logs (location specified during install)
+---bin
|   +---optimize
|   +---debug
|   \---config -> ../config
+---config
|   +---internal
|   +---mibs
|   \---snapshots
+---ui
|   +---bb
|   +---images
|   \---netcharts
|     +---graphics
|      +---gui
|      \---util
```

Traffic Server Binaries

The Traffic Server **bin** directory contains the binary executables and scripts used during operation of the traffic server. It also contains both optimized and debug versions of the binaries in the **optimize** and **debug** subdirectories respectively. By default the optimized binaries are the ones utilized in the **bin** directory and this should not be changed unless directed by Inktomi support. There is also a symbolic link to the **config** directory that should not be removed.

Executables of interest to Traffic Server users include:

example_alarm_bin.sh – This shell script is used to send email to the Traffic Server administrator when the server experiences an alarm. The administrator's email address must be set here.

start_traffic_server – This shell script is used to start the Traffic Server processes and should be executed by the Unix user the Traffic Server is configured to run as (or root).

stop_traffic_server – This shell script is used to stop the Traffic Server processes and should be executed by the Unix user the Traffic Server is configured to run as (or root).

traffic_line – Command line interface to the Traffic Manager

```
> ./traffic_line -h
Usage: ./traffic_line [--SWITCH [ARG]]
  switch_____________________type__default___description
  -i, --interactive       on    false     Interactive Mode
  -p, --socket_path       str   ./confi.. Socket Path
  -r, --read_var          str   (null)    Read Varaible
  -s, --set_var           str   (null)    Set Varialble(requires -v option)
  -v, --value             str   (null)    Set Value(used with -s option)
  -h, --help                              Help
  -x, --reread_config     on    false     Reread Config Files
  -M, --restart_cluster   on    false     Restart traffic_manager (cluster wide)
  -L, --restart_local     on    false     Restart traffic_manager (local node)
  -S, --shutdown          on    false     Shutdown traffic_server (local node)
  -U, --startup           on    false     Start traffic_server (local node)
  -B, --bounce_cluster    on    false     Bounce traffic_server (cluster wide)
  -b, --bounce_local      on    false     Bounce local traffic_server
  -T, --timeout           int   -1        Request timeout (seconds)

> ./traffic_line -i cli-
> help
    1.  monitor           # monitor mode
    2.  configure         # configure mode
    3.  reread            # forces a reread of the configuration files
    4.  shutdown          # Shuts down the traffic_server
    5.  startup           # Starts the traffic_server (local node)
    6.  bounce_local      # Restarts the traffic_server (local node)
    7.  bounce_cluster    # Restarts the traffic_server (cluster wide)
    8.  restart_local     # Restarts the traffic_manager (local node)
    9.  restart_cluster   # Restarts the traffic_manager (cluster wide)
    Select above options by number

    help                  # displays a list of commands
```

traffic_mom.tab – sample crontab file for the Traffic Server user to oversee traffic_cop is running and execute the traffic_cop binary if not every five minutes.

traffic_cop, traffic_manager, traffic_server – The main Traffic Server binaries.

Traffic Server Configuration Files

The Traffic Server **config** directory holds the configuration files used by the Traffic Server. The configuration files of interest to a Traffic Server user are mainly created and modified by the installation script and the Traffic Manager GUI.

The **config** directory contains an **internal** subdirectory which contains configuration files used by the Traffic Server processes to identify the current pid of each running Traffic Server process (***.lock** files). If the Traffic Server is not running then a **no_cop** file will be present instead. Additional files in this directory indicate the current configuration of the on-disk object store and host cache. The files in this directory should not be modified!

The **config** directory contains a **snapshots** subdirectory which contains backup copies of configuration files at the user's discretion. The user can record a "snapshot" of the current configuration at any time through the Traffic Manager GUI. It also contains the **mibs** directory (for SNMP).

Configuration files of interest to the Traffic Server user are commented with usage and syntax information. Remember that changes to a single configuration will be propagated to all nodes in a clustered environment.

S97traffic_server – a shell script for starting the Traffic Server at boot time. Placed in /etc/rc2.d by the installation script. Sets kernel and TCP tunable values, prepares Traffic Server environment and executes the traffic_cop binary to start the Traffic Server.

cache.config – The Traffic Server caches objects indexed by URLs. In this configuration file, you can specify how a particular group of URLs should be cached. Rules include:

- Whether to cache objects
- How long to pin particular objects in the cache
- How long to consider cached objects still fresh

After you modify cache.config, the Traffic Manager has to reread te configuarion files. To do this move to the /bin directory and run the traffic_line command utility with the -x command. A sample change might be: **`dest_domain=112.12.12.12 scheme=ftp action=never-cache`**. Syntax is:
<primary destination>=value <secondary specifier>=value <action>=value

```
#$Id: cache.config,v 1.13 1998/08/19 17:39:38 peter Exp $
# cache.config
#
# The purpose of this file is to alter caching parameters of
#   specific objects or sets of objects
#
# Each line consists of a set of tag value pairs.  The pairs
#   are in the format  <tag>=<value>
#
# Each line must include exactly one primary specifier
#
#   Primary destination specifiers are
#     dest_domain=         (Requested IP address or domain name)
#     dest_host=                 (Requested IP address or domain name)
#     dest_ip=                   (Requested IP address)
#     url_regex=                 (Regular expression to be found in URL)
#
#
# Lines may include any number of the secondary specifiers but
#    secondary specifiers may not be duplicated on the same line
#
#   Secondary specifiers are
#     port=                      (A requested URL port)
#     scheme=              (A requested URL proocol - HTTP or FTP)
#     prefix=              (A prefix in the path of a URL)
#     suffix=              (A file suffix in the URL)
#     method=              (A request URL method - get, post, put, face)
#     time=                      (A time range - 08:00 - 14:00)
#     src_ip=              (IP address of the client)
#
# Each Lines must include a exactly one cache cache directive
#   Cache directives are
#     action=never-cache   (To origin server without caching)
```

```
#     action=ignore-no-cache     (To origin server without caching)
#     pin-in-cache=<time>  (Amount of time to pin object in cache 1h15m20s)
#     revalidate=<time>          (As above - specify hours, minutes, seconds)
#
#
# Examples
#
#  Revalidate all http objects from inktomi.com after 2 hours
#    dest_domain=inktomi.com   scheme=http  revalidate=2h
#
#  Revalidate all ftp objects from inktomi.com after 2 days
#    dest_domain=inktomi.com   scheme=ftp  revalidate=2d
#
#
```

filter.config – Allows you to deny or allow particular URL requests and keep or strip header information. Allowing a URL means that the Traffic Server will cache and serve the requested document. Denying a URL means that requests will be forwarded to the origin server. After you modify filter.config, the Traffic Manager has to reread te configuarion files. To do this move to the /bin directory and run the traffic_line command utility with the -x command. A sample change might be: **`dest_domain=112.12.12.12 scheme=ftp action=deny`**. Syntax is: <primary destination>=value <secondary specifier>=value <action>=value.

```
#$Id: filter.config,v 1.2 1998/07/02 17:57:49 mchowla Exp $
# filter.config
#
# The purpose of this file is to specify which http and ftp
#  objects can be obtained through Traffic Server and which
#  headers should be forwarded for http requests
#
# Each line consists of a set of tag value pairs.  The pairs
#   are in the format  <tag>=<value>
#
# Each line must include exactly one primary specifier
#
#   Primary destination specifiers are
#     dest_domain=         (Requested IP address or domain name)
#     dest_host=                 (Requested IP address or domain name)
#     dest_ip=                   (Requested IP address)
#     url_regex=                 (Regular expression to be found in URL)
#
#
# Lines may include any number of the secondary specifiers but
#    secondary specifiers may not be duplicated on the same line
#
#   Secondary specifiers are
#     port=                      (A requested URL port)
#     scheme=              (A requested URL protocol - HTTP or FTP)
#     prefix=              (A prefix in the path of a URL)
#     suffix=              (A file suffix in the URL)
#     method=              (A request URL method - get, post, put, face)
#     time=                      (A time range - 08:00 - 14:00)
#     src_ip=              (IP address of the client)
#
#
# Each Lines must include a exactly one action
#   Actions are
#     action=allow         (Cache the selected URL)
#     action=deny                (Do not cache the selected URL)
#     keep_hdr=<hdr Name>  (Header info to keep: date, host, cookie, client_ip)
#     strip_hdr=<hdr Name> (Header info to strip: same as keep_hdr)
#
```

```
# Note: in the case of conflicting directives, the directive
#   that appears first applies
#
# Examples:
#
#   The only host that is accessible in the domain inktomi.com is
#     www.inktomi.com
#   dest_host=www.inktomi.com     action=allow
#   dest_domain=inktomi.com       action=deny
#   url_regex=politics prefix=/viewpoints keep_hdr=client_ip
```

icp.config – Used to define ICP peers (parent and sibling caches). Each line in the icp.config file contains the name and configuration information for a single ICP peer. Syntax is shown below in the file:

```
#$Id: icp.config,v 1.9 1998/08/26 06:03:18 davey Exp $
#
# $Id: icp.config,v 1.9 1998/08/26 06:03:18 davey Exp $
##############################################################################
#
# ICP Configuration -- Defines ICP parent/sibling configuration
#
#  Each line is formatted as follows with ":" separator for each field.
#    - hostname (string)           -- Identifier for entry
#    - host_ip_str (string)        -- decimal dot notation
#    - ctype (int)                 -- 1=Parent, 2=Sibling
#    - proxy_port (int)            -- TCP Port #
#    - icp_port (int)              -- UDP Port #
#    - multicast_member            -- 0=No 1=Yes
#    - multicast_ip_str (string)   -- decimal dot notation
#                                  224.0.0.0 - 239.255.255.255
#    - multicast_ttl (int)         -- (1 - 2; default 1)
#
# <host>:<host IP>:<ctype>:<proxy port>:<icp port>:<MC on>:<mc IP>:<MC ttl>:
#
# Example #1 (1 parent and 1 sibling):
# ============================================================
#     host1:209.1.33.10:1:8080:3130:0:0.0.0.0:0:
#     host2:209.1.33.11:2:8080:3130:0:0.0.0.0:0:
#
# Example #2 (1 parent and 1 sibling using MultiCast):
# ============================================================================
#     host1:209.1.33.10:1:8080:3130:1:239.128.16.128:1:
#     host2:209.1.33.11:2:8080:3130:1:239.128.16.128:1:
#
##############################################################################
```

ip_allow.config – Used to specify clients allowed to access the Traffic Server. You can specify ranges of IP addresses that are allowed to use the Traffic Server as a web proxy. If you want to deny Traffic Server access to specific IP addresses, do not include them in any line in the ip_allow.config file. Be sure to re-read the configuration file with traffic_line -x.

```
#$Id: ip_allow.config,v 1.2 1998/04/09 21:55:21 mchowla Exp $
# ip_allow.config
#
#
#
src_ip=0.0.0.0-255.255.255.255          action=ip_allow
```

logs.config – Used to specify a custom log file format. You can enable or disable standard format access logs in records.config or (preferably) through the Traffic Manager UI. Be sure to re-read the configuration file with traffic_line -x.

```
# logs.config
#
# $Id: logs.config,v 1.19 1998/07/07 00:32:01 jonb Exp $
# Copyright 1997 Inktomi Corporation. All Rights Reserved.  Confidential.
#
# This is the configuration file for Traffic Server logging.  The purpose
# of this file is to define custom logging formats and filters.  Standard
# event log formats (Squid, Netscape Common, Netscape Extended, Netscape
# Extended2) are built into the Traffic Server logging system and can be
# enabled/disabled from the logging configuration user interface.
#
# NOTE: Because custom log formats are completely dynamic, there is a
# possible performance impact for using them.  For optimal logging
# performance, use one of the pre-defined logging formats.
#
# An event log format specifies which fields are to be gathered from each
# Http/ICP/NNTP access event and placed into the log file as a log entry.
# For each new custom log format type, the following information is needed:
#
#   the word "format"
#   enabled or disabled
#   unique format identifier integer
#   format name
#   printf-style format string specifying the field symbols and how they
#      should look in ASCII
#   file name
#   file type: either ASCII or BINARY
#   file header data (or "none")
#
# All of this information is placed on a SINGLE line, following the
# identifier "format" and separated by colons (':').  Example:
#
# format:enabled:1:minimal:%<chi> / %<cqu> / %<pssc>:minimal:ASCII:none
#
# One final note: the code that parses this is fairly simple-minded (like
# its designer), so make sure you follow all of the rules exactly.
```

mgmt_allow.config – Used to specify hosts allowed to access the Traffic Manager UI. If you want to deny Traffic Manager access to specific IP addresses, do not include them in any line in this file. Be sure to re-read the configuration file with traffic_line -x.

```
#$Id: mgmt_allow.config,v 1.1 1998/04/09 21:54:05 mchowla Exp $
# mgmt_allow.config
#
#
#
src_ip=0.0.0.0-255.255.255.255          action=ip_allow
```

nntp_access.config – Used to specify the access privileges for a particular group of NNTP clients. Each line describes the access privileges for a particular group of clients. Be sure to re-read the configuration file with traffic_line -x.

```
# nntp_access.config
#
# There are three ways of specifying groups of clients:
# by IP range, domain and hostname.
#
# Examples:
#
# ip = 0.0.0.0-255.255.255.255
```

```
# ip = 127.0.0.1
# domain = inktomi.com
# hostname = myhost.mydomain.com
#
#  ip=127.0.0.1 acces="generic" authenticator="homebrew" user="joe"
#  ip=127.0.0.1 acces="custom" authenticator="hb" user=required pass=required
#
# For each group of clients an access directive can be given.  The directives
# are "allow", "deny", "basic", "generic", and "custom".
#
# For 'basic', the "user" option is required and "pass" is optional.
#
# hostname = myhost.mydomain.com access="basic" user="joe" pass="bob"
#
# For 'generic', the "authenticator" option is optional.
#
# For 'custom', the "authenticator" option is requred.  The "user" and
# "pass" options are optional, but if present must be the string 'required'
# indicating that the user and password will be required of the user and
# then passed to the "authenticator" program through the "generic" interface.
#
# hostname = myhost.mydomain.com access="custom" authenticator = "homebrew" user =
#  required pass = required
#
# Directives that appear earlier in the file take precedence.
#
ip=0.0.0.0-255.255.255.255 access=allow
```

nntp_servers.config - Use this file to configure the Traffic Server's parent NNTP servers, the news groups the Traffic Server observes, NNTP activity types and the network interface Traffic Server should use to contact the parent NNTP. Be sure to re-read the configuration file with traffic_line -x.

```
# NNTP configuration
#
#$Id: nntp_servers.config,v 1.7 1998/07/21 21:52:15 jplevyak Exp $
#
# The format of this file is a sequence of lines:
#
#   hostname <group-wildmat> (<priority>) (<interface>)
#
# which describe the upstream servers for particular groups.
#
# hostname is a hostname or IP and option port.  The special token
# ".block" means block access to these newsgroups.
#
#   Examples:
#       localhost:120
#       qqqqq.com
#       10.100.34.1:9999
#       10.2.2.1
#       .block
#
# <group-wildmat> is a comma separated list of groupnames and 'list files'
# in 'wildmat' format. There can be *no spaces* in the list.  Groups
# which should not be included can be prefixed with a '!'.  Note: for
# compatibility with INN, the list is processed in *reverse* order and
# action is taken on the *last* match, so more specific restrictions/allows
# should be placed *later* in the list.  The 'list files' are 'subscriptions',
# 'distributions' and 'distrib.pats'.  The 'active', 'active.times' and
# 'newsgroups' files are all cached from all servers.
#
#   Examples:
#       comp.compilers
```

```
#       talk.religion.*,!talk.religion.barney,subscriptions
#       alt.*,rec.*,soc.*
#       *,!alt.*
#       *,!distrib.pats
#
# (<priority>) is an optional priority, feed or post designator.
# The exclusive feed designator is the token 'feed' which
# prevents Traffic Server from attempting to retrieve news on demand.
# The non-exclusive feed designator is "push" (allow push traffic) which
# allows a partial or full feed to be combined with demand retrieval.
# The token "pull" tells the Traffic Server to actively pull news
# instead of waiting for a user request.  Likewise, "pullover" causes
# the Traffic Server to actively pull the overview database while
# retrieving the articles on demand.  The "dynamic" designation
# causes the Traffic Server to automatically decide whether a group
# should be "pull", "pullover" or demand retrieval based on the usage pattern.
#
# The token "post" indicates that articles posted to the designated
# groups should be sent to the particular server.  The first matching
# server for the first matching group is used.  If no "post" hosts match,
# the first matching highest priority host is used.
#
# For non-feeds, when news from a particular group is required,
# a matching server with the lowest priority is used.  If there are
# multiple servers at the same priority, they are accessed in a round-robin
# fashion.  The default priority is 0.
#
# Hosts can appear as "feed", "push", "pull", "pullover", "dynamic", "post"
# and non-feed hosts.
#
#   Examples:
#       news.backup.qqqq.com     rec.*  push
#       news.backup.qqqq.com     soc.*  dynamic
#       postnews.qqq.com        *       post
#       alt.qqqq.com             alt.*  1
#       alt2.qqqq.com:9999 alt.*  1
#       news.qqqq.com           *       2
#       10.2.2.1          *      3
#       news.backup.qqqq.com     *      4
#
# (<interface>) is the network interface which should be used to
# contact the remote host.
#
#   Examples:
#       .block                  !rec.soccer,rec.*
#       comp.qqqq.com:9999 comp.* feed   10.3.3.2
#       alt2.qqqq.com:9999 alt.*  1      10.3.3.2
#       news.backup.qqqq.com     *      3      secondary.qqqq.com
#
# NOTE: some server should be given which supplies the group '.nogroup' for
#       those commands which do not specify a group (article <message-id).
#       This could be a server which supplies *.
#   Examples:
#        block                  !rec.soccer,rec.*
#       comp.qqqq.com:9999 comp.*,.nogroup,!microsoft.*
#       msnews.microsoft.com.   microsoft.*
#
#msnews.microsoft.com.   microsoft.*
#news.mozilla.org.  netscape.*,people.*
```

parent.config – Used to specify HTTP parent proxy hierarchies and selected URL requests to by pass parent proxies. Be sure to re-read the configuration file with traffic_line -x.

```
#$Id: parent.config,v 1.2 1998/07/02 18:03:45 mchowla Exp $
# parent.config
```

```
#
# The purpose of this file is to specify the parent proxy for
#   specific objects or sets of objects
#
# Each line consists of a set of tag value pairs.  The pairs are in the format
# <tag>=<value>
#
# Each line must include exactly one primary specifier
#   Primary destination specifiers are
#     dest_domain=
#     dest_host=
#     dest_ip=
#     url_regex=
#
# Lines may include any number of the secondary specifiers but
#    secondary specifiers may not be duplicated on the same line
#
#   Secondary specifiers are
#     port=
#     scheme=
#     prefix=
#     suffix=
#     method=
#     time=
#     src_ip=
#
# Available parent directives are:
#     parent=    (a semicolon separated list of parent proxyies)
#     go_direct={true,false}
#     round_robin={true,false}
#
# Each line must include a parent= directive or a go_direct=
#   directive.  If both appear, Traffic Server will directly
#   contact the origin server if all the listed parent proxies are down
#
# Example
#
#  Alternate requests between proxy1 and proxy2
#
# dest_domain=.  parent="proxy1.inktomi.com:8080; proxy2.inktomi.com:8080"
# round_robin=true
```

proxy.pac – Proxy Auto Configuration file. Can be configured through the Traffic Manager GUI or manually edited once you have indicated you want to set up an auto-configuration file.

```
function FindProxyForURL(url, host) {

        // Make sure this a protocol we proxy
        if(!((url.substring(0,5) == "http:") ||
                (url.substring(0,4) == "ftp:") ||
                (url.substring(0,6) == "https:"))) {
                 return "DIRECT";
        }
        return "PROXY ink-proxy.inktomi.com:8090;" +
               "PROXY proxydev.inktomi.com:8090;" +
               "DIRECT";
        }
```

records.config – The main Traffic Server configuration file containing most of the configuration variables for both the Traffic Manager and Traffic Server. Modified extensively through the Traffic Manager GUI or manually as directed by support. Proceed cautiously because many variables are "coupled," meaning they interact with other variables. Changing a single variable in isolation could cause the Traffic Server to fail.

It is best to use the Traffic Manager UI or Traffic Line (command line interface) to make changes in this file. Be sure to re-read the configuration file with traffic_line -x.

```
#$Id: records.config,v 1.289.2.81 1998/11/03 05:25:35 haines Exp $
#
# Process Records Config File
#
# <RECORD-TYPE> <NAME> <TYPE> <VALUE (till end of line)>
#
#      RECORD-TYPE:  CONFIG or PROCESS(stat)
#      NAME:         name of variable
#      TYPE:         INT, COUNTER, STRING, FLOAT
#      VALUE:        Initial value for record
#
###############################################################################
#
# System Variables
#
###############################################################################
CONFIG proxy.config.proxy_name STRING ts-sgi2
CONFIG proxy.config.bin_path STRING bin
CONFIG proxy.config.proxy_binary STRING traffic_server
CONFIG proxy.config.proxy_binary_opts STRING -M
CONFIG proxy.config.manager_binary STRING traffic_manager
CONFIG proxy.config.cli_binary STRING traffic_line
CONFIG proxy.config.watch_script STRING traffic_cop
CONFIG proxy.config.start_script STRING start
CONFIG proxy.config.env_prep STRING example_prep.sh
CONFIG proxy.config.config_dir STRING config
CONFIG proxy.config.temp_dir STRING /tmp
CONFIG proxy.config.alarm_email STRING inktomi
CONFIG proxy.config.syslog_facility STRING LOG_DAEMON
# Negative core limit means max out limit
CONFIG proxy.config.core_limit INT 0
# 0 = disable (seconds)
CONFIG proxy.config.dump_mem_info_frequency INT 0
###############################################################################
#
# Version Info
#
###############################################################################
PROCESS proxy.process.version.server.short STRING NULL
PROCESS proxy.process.version.server.long STRING NULL
PROCESS proxy.process.version.server.build_number STRING NULL
PROCESS proxy.process.version.server.build_time STRING NULL
PROCESS proxy.process.version.server.build_date STRING NULL
PROCESS proxy.process.version.server.build_machine STRING NULL
PROCESS proxy.process.version.server.build_person STRING NULL
###############################################################################
#
# Diagnostics
#
# Enable by setting proxy.config.diags.debug.enabled to 1
# Route each type of diagnostic with a string, each character representing:
#    O  stdout
#    E  stderr
#    S  syslog
#    L  diags.log
###############################################################################
CONFIG proxy.config.diags.debug.enabled INT 0
CONFIG proxy.config.diags.debug.tags STRING NULL
CONFIG proxy.config.diags.action.enabled INT 0
CONFIG proxy.config.diags.action.tags STRING NULL
CONFIG proxy.config.diags.show_location INT 0
```

```
CONFIG proxy.config.diags.output.diag STRING E
CONFIG proxy.config.diags.output.debug STRING E
CONFIG proxy.config.diags.output.status STRING S
CONFIG proxy.config.diags.output.note STRING S
CONFIG proxy.config.diags.output.warning STRING S
CONFIG proxy.config.diags.output.error STRING SE
CONFIG proxy.config.diags.output.fatal STRING SE
CONFIG proxy.config.diags.output.alert STRING SE
CONFIG proxy.config.diags.output.emergency STRING SE
##############################################################################
#
# Local Manager
#
##############################################################################
CONFIG proxy.config.lm.pserver_timeout_secs INT 1
CONFIG proxy.config.lm.pserver_timeout_msecs INT 0
CONFIG proxy.config.lm.sem_id INT 11452
CONFIG proxy.config.cluster.delta_thresh INT 30
CONFIG proxy.config.cluster.peer_timeout INT 30
CONFIG proxy.config.cluster.startup_timeout INT 10
# cluster type requires restart to change
# 1 is full clustering, 2 is mgmt only, 3 is no clustering
CONFIG proxy.config.cluster.type INT 3
CONFIG proxy.config.cluster.rsport INT 8088
CONFIG proxy.config.cluster.mcport INT 8089
CONFIG proxy.config.cluster.mc_group_addr STRING 224.0.1.37
CONFIG proxy.config.admin.html_doc_root STRING ui
CONFIG proxy.config.admin.web_interface_port INT 8081
CONFIG proxy.config.admin.dynamic_graph_port INT 8082
CONFIG proxy.config.admin.autoconf_port INT 8083
CONFIG proxy.config.admin.overseer_port INT -1
CONFIG proxy.config.admin.admin_user STRING admin
CONFIG proxy.config.admin.admin_password STRING admin
CONFIG proxy.config.admin.guest_user STRING admin
CONFIG proxy.config.admin.guest_password STRING admin
CONFIG proxy.config.feature_set INT 1
CONFIG proxy.config.admin.basic_auth INT 1
CONFIG proxy.config.admin.use_ssl INT 0
CONFIG proxy.config.admin.ssl_cert_file STRING private_key.pem
CONFIG proxy.config.admin.number_config_bak INT 3
CONFIG proxy.config.admin.user_id STRING inktomi
CONFIG proxy.config.admin.ui_refresh_rate INT 30
CONFIG proxy.config.admin.load_factor FLOAT 250.00
CONFIG proxy.config.admin.log_mgmt_access INT 0
CONFIG proxy.config.admin.log_resolve_hostname INT 1
CONFIG proxy.config.admin.ip_allow.filename STRING mgmt_allow.config
CONFIG proxy.config.admin.advanced_ui INT 1
##############################################################################
#
# Process Manager
#
##############################################################################
CONFIG proxy.config.process_manager.timeout INT 5
CONFIG proxy.config.process_manager.enable_mgmt_port INT 1
CONFIG proxy.config.process_manager.mgmt_port INT 8084
##############################################################################


# Virtual IP Manager
#
##############################################################################
CONFIG proxy.config.vmap.enabled INT 0
CONFIG proxy.config.vmap.addr_file STRING vaddrs.config
CONFIG proxy.config.vmap.down_up_timeout INT 10
CONFIG proxy.config.ping.npacks_to_trans INT 5
```

```
CONFIG proxy.config.ping.timeout_sec INT 1
##############################################################################
#
# Alarm Configuration
#
##############################################################################
       ###################################################################
       # execute alarm as "<abs_path>/<bin> "<MSG_STRING_FROM_PROXY>"" #
       ###################################################################
CONFIG proxy.config.alarm.bin STRING example_alarm_bin.sh
CONFIG proxy.config.alarm.abs_path STRING NULL
##############################################################################
#
# Transparency Configuration
#
##############################################################################
CONFIG proxy.config.trans.enabled INT 0
CONFIG proxy.config.trans.nat_config_file STRING /usr/people/inktomi/trans/config/ipnat.conf

CONFIG proxy.config.trans.acl_filename STRING NULL
CONFIG proxy.config.trans.bypass_enabled INT 0
CONFIG proxy.config.trans.bypass_use_and_rules INT 0

CONFIG proxy.config.trans.bypass_use_and_rules_bad_client_request INT 0
CONFIG proxy.config.trans.bypass_use_and_rules_400 INT 0
CONFIG proxy.config.trans.bypass_use_and_rules_401 INT 0
CONFIG proxy.config.trans.bypass_use_and_rules_403 INT 0
CONFIG proxy.config.trans.bypass_use_and_rules_405 INT 0
CONFIG proxy.config.trans.bypass_use_and_rules_406 INT 0
CONFIG proxy.config.trans.bypass_use_and_rules_408 INT 0
CONFIG proxy.config.trans.bypass_use_and_rules_500 INT 0

CONFIG proxy.config.trans.bypass_on_bad_client_request INT 0
CONFIG proxy.config.trans.bypass_on_400 INT 0
CONFIG proxy.config.trans.bypass_on_401 INT 0
CONFIG proxy.config.trans.bypass_on_403 INT 0
CONFIG proxy.config.trans.bypass_on_405 INT 0
CONFIG proxy.config.trans.bypass_on_406 INT 0
CONFIG proxy.config.trans.bypass_on_408 INT 0
CONFIG proxy.config.trans.bypass_on_500 INT 0

PROCESS proxy.process.trans.num_bypass_on_bad_client_request INT 0
PROCESS proxy.process.trans.num_bypass_on_400 INT 0
PROCESS proxy.process.trans.num_bypass_on_401 INT 0
PROCESS proxy.process.trans.num_bypass_on_403 INT 0
PROCESS proxy.process.trans.num_bypass_on_405 INT 0
PROCESS proxy.process.trans.num_bypass_on_406 INT 0
PROCESS proxy.process.trans.num_bypass_on_408 INT 0
PROCESS proxy.process.trans.num_bypass_on_500 INT 0

        ############
        # Parsing #
        ############
# Should we use the Host: header field to determine the origin server
# if the request URL does not contain the hostname? This happens when
# transparent proxying is being done.
CONFIG proxy.config.header.parse.use_host_header_field INT 1
CONFIG proxy.config.header.parse.no_host_url_redirect STRING NULL
##############################################################################
#
# Inktoswitch Configuration
#
##############################################################################
CONFIG proxy.config.http.inktoswitch_enabled INT 0
CONFIG proxy.config.http.router_ip INT 0
```

```
CONFIG proxy.config.http.router_port INT 0
##############################################################################
#
# Transform Configuration
#
##############################################################################
CONFIG proxy.config.transform.jg_compd_port INT 1815
CONFIG proxy.config.transform.jg_compd_addrs STRING NULL
CONFIG proxy.config.transform.jg_compress_no_content_length INT 1
CONFIG proxy.config.transform.jg_min_content_length INT 4
CONFIG proxy.config.transform.jg_max_content_length INT 8192
CONFIG proxy.config.transform.jg_perform_status_check INT 1
CONFIG proxy.config.transform.jg_retry_thresh INT 0
##############################################################################
#
# HTTP Engine
#
##############################################################################
       ##########
       # basics #
       ##########
       #
       # The main server_port is listed here, other server ports is a
       # string of ports, separated by whitespace.
       #
       # The port attributes should be set to X(default behavior). For
       # example ...server_other_ports STRING 1234:X 12345:X
       #
CONFIG proxy.config.http.server_port INT 8080
CONFIG proxy.config.http.server_port_attr STRING X
CONFIG proxy.config.http.server_other_ports STRING NULL
CONFIG proxy.config.http.request_via_str STRING Traffic-Server/2.0
CONFIG proxy.config.http.response_via_str STRING Traffic-Server/2.0
CONFIG proxy.config.http.user_language INT 1
CONFIG proxy.config.http.enable_url_expandomatic INT 1
CONFIG proxy.config.http.keep_alive_enabled INT 1
CONFIG proxy.config.http.verbose_via INT 1
       #  origin_server_pipeline and user_agent_pipeline
       #
       #  0      - no keepalive
       #  n >= 1 - max pipeline window
       #           (1 is the same HTTP/1.0 keepalive)
       #
CONFIG proxy.config.http.origin_server_pipeline INT 1
CONFIG proxy.config.http.user_agent_pipeline INT 4
CONFIG proxy.config.http.ftp_enabled INT 1
CONFIG proxy.config.http.wuts_enabled INT 0
CONFIG proxy.config.http.record_heartbeat INT 1
       ###############################
       # parent proxy configuration #
       ###############################
CONFIG proxy.config.http.ssl_parent_proxy STRING NULL
CONFIG proxy.config.http.ssl_parent_proxy_port INT 8080
CONFIG proxy.config.http.parent_proxy_routing_enable INT 0
CONFIG proxy.config.http.parent_proxies STRING NULL
CONFIG proxy.config.http.parent_proxy.file STRING parent.config
CONFIG proxy.config.http.parent_proxy.retry_time INT 300
       ###################################
       # HTTP connection timeouts (secs) #
       ###################################
       #
       # out: proxy -> os connection
       # in : ua -> proxy connection
       #
CONFIG proxy.config.http.keep_alive_no_activity_timeout_in INT 10
```

```
CONFIG proxy.config.http.keep_alive_no_activity_timeout_out INT 10
CONFIG proxy.config.http.transaction_no_activity_timeout_in INT 120
CONFIG proxy.config.http.transaction_no_activity_timeout_out INT 120
CONFIG proxy.config.http.transaction_active_timeout_in INT 1800
CONFIG proxy.config.http.transaction_active_timeout_out INT 1800
CONFIG proxy.config.http.accept_no_activity_timeout INT 120
CONFIG proxy.config.http.background_fill_active_timeout INT 10
CONFIG proxy.config.http.background_fill_no_activity_timeout INT 10
       ###################################
       # origin server connect attempts #
       ###################################
CONFIG proxy.config.http.connect_attempts_max_retries INT 6
CONFIG proxy.config.http.connect_attempts_timeout INT 30
CONFIG proxy.config.http.down_server.cache_time INT 900
CONFIG proxy.config.http.down_server.abort_threshold INT 10
      ##########################
      # proxy users variables #
      ##########################
CONFIG proxy.config.http.anonymize_remove_from INT 0
CONFIG proxy.config.http.anonymize_remove_referer INT 0
CONFIG proxy.config.http.anonymize_remove_user_agent INT 0
CONFIG proxy.config.http.anonymize_remove_cookie INT 0
CONFIG proxy.config.http.anonymize_remove_client_ip INT 0
CONFIG proxy.config.http.anonymize_insert_client_ip INT 1
CONFIG proxy.config.http.anonymize_other_header_list STRING NULL
       ##########
       # attack #
       ##########
CONFIG proxy.config.http.request_header_max_size INT 16384
CONFIG proxy.config.http.response_header_max_size INT 16384
      #################
      # cache control #
      #################
CONFIG proxy.config.http.cache.on INT 1
CONFIG proxy.config.http.cache.http INT 1
CONFIG proxy.config.http.cache.ftp INT 1
CONFIG proxy.config.http.cache.ignore_client_no_cache INT 0
CONFIG proxy.config.http.cache.cache_responses_to_cookies INT 1
CONFIG proxy.config.http.cache.cache_urls_that_look_dynamic INT 0
CONFIG proxy.config.http.cache.enable_default_vary_headers INT 1
CONFIG proxy.config.http.cache.compress_vary_default_opt INT 0
      #  when_to_revalidate has 4 options:
      #
      #  0 - default. use use cache directives or heuristic
      #  1 - stale if heuristic
      #  2 - always stale (always revalidate)
      #  3 - never stale
      #
CONFIG proxy.config.http.cache.when_to_revalidate INT 0
      #  required headers: three options
      #
      #  0 - No required headers to make document cachable
      #  1 - at least, "Last-Modified:" header required
      #  2 - explicit lifetime required, "Expires:" or "Cache-Control:"
      #
CONFIG proxy.config.http.cache.required_headers INT 0
CONFIG proxy.config.http.cache.add_content_length INT 0
      ##########################
      # heuristic expiration #
      ##########################
CONFIG proxy.config.http.cache.heuristic_min_lifetime INT 3600
CONFIG proxy.config.http.cache.heuristic_max_lifetime INT 86400
CONFIG proxy.config.http.cache.heuristic_lm_factor FLOAT 0.10
CONFIG proxy.config.http.cache.guaranteed_min_lifetime INT 0
CONFIG proxy.config.http.cache.guaranteed_max_lifetime INT 604800
```

```
      ########################################
      # dynamic content & content negotiation #
      ########################################
CONFIG proxy.config.http.cache.vary_default_text STRING Cookie
CONFIG proxy.config.http.cache.vary_default_images STRING NULL
CONFIG proxy.config.http.cache.vary_default_other STRING NULL
      ##########################
      # anonymous ftp password #
      ##########################
CONFIG proxy.config.http.ftp.anonymous_passwd STRING inktomi

       ################################
       # cached ftp document lifetime #
       ################################
CONFIG proxy.config.http.ftp.cache.document_lifetime INT 259200

       ###################
       # Error Reporting #
       ###################
CONFIG proxy.config.http.errors.log_error_pages INT 1

###############################################################################
#
# Customizable Response Pages
#
###############################################################################
CONFIG proxy.config.body_factory.enabled INT 1
CONFIG proxy.config.body_factory.logging_enabled INT 1
CONFIG proxy.config.body_factory.template_dir STRING config/body_factory

###############################################################################
#
# NNTP Engine
#
###############################################################################
CONFIG proxy.config.nntp.enabled INT 0
CONFIG proxy.config.nntp.cache_enabled INT 1
CONFIG proxy.config.nntp.posting_enabled INT 1
CONFIG proxy.config.nntp.access_control_enabled INT 0
CONFIG proxy.config.nntp.v2_authentication INT 0
CONFIG proxy.config.nntp.cluster_enabled INT 1
CONFIG proxy.config.nntp.feed_enabled INT 1
CONFIG proxy.config.nntp.logging_enabled INT 1
CONFIG proxy.config.nntp.background_posting_enabled INT 0
CONFIG proxy.config.nntp.insert_posting_trace_header INT 1
CONFIG proxy.config.nntp.posting_ok_message STRING Inktomi NNTP server ready. posting ok
CONFIG proxy.config.nntp.posting_not_ok_message STRING Inktomi NNTP server ready. no posting
CONFIG proxy.config.nntp.servers_filename STRING nntp_servers.config
CONFIG proxy.config.nntp.access_filename STRING nntp_access.config
CONFIG proxy.config.nntp.server_port INT 119
# null defaults to localhost
CONFIG proxy.config.nntp.authorization_hostname STRING NULL
CONFIG proxy.config.nntp.authorization_port INT 0
CONFIG proxy.config.nntp.obey_control_cancel INT 0
CONFIG proxy.config.nntp.obey_control_newgroup INT 0
CONFIG proxy.config.nntp.obey_control_rmgroup INT 0
# in seconds, must be at least 3 minutes as per draft standard
CONFIG proxy.config.nntp.inactivity_timeout INT 600
# all in seconds 86400 = day, 3600 = hour, 60 = minute
CONFIG proxy.config.nntp.check_newgroups_every INT 86400
CONFIG proxy.config.nntp.check_newnews_every INT 900
CONFIG proxy.config.nntp.check_cancels_every INT 3600
CONFIG proxy.config.nntp.maintain_every INT 120
CONFIG proxy.config.nntp.check_pull_every INT 600
CONFIG proxy.config.nntp.group_check_parent_every INT 300
```

```
CONFIG proxy.config.nntp.group_check_cluster_every INT 60
CONFIG proxy.config.nntp.group_sync_every INT 600
CONFIG proxy.config.nntp.group_expire_every INT 28800
CONFIG proxy.config.nntp.overview_sync_every INT 120
CONFIG proxy.config.nntp.overview_gc_every INT 1200
CONFIG proxy.config.nntp.load_overview_min INT 25
CONFIG proxy.config.nntp.server_retry_timeout INT 60
# throttle in bytes per second.  0 = no throttle
CONFIG proxy.config.nntp.client_speed_throttle INT 0
CONFIG proxy.config.nntp.max_articles_per_group INT 100000
# NNTP Authentication Server Config
# in milliseconds
CONFIG proxy.config.nntp.auth_server.binary STRING nntp_auth
CONFIG proxy.config.nntp.run_local_authentication_server INT 0
CONFIG proxy.config.nntp.accept_local_authentication_requests_only INT 1
# currently not supported
CONFIG proxy.config.nntp.custom_authentication_via_stdio INT 0
CONFIG proxy.config.nntp.authorization_server_timeout INT 50000
# whether or not nntp status information is available at http://{nntp}
CONFIG proxy.config.nntp.scope INT 1

# Statistics
PROCESS proxy.process.nntp.client_connections_currently_open INT 0
PROCESS proxy.process.nntp.client_bytes_read INT 0
PROCESS proxy.process.nntp.client_bytes_written INT 0
PROCESS proxy.process.nntp.server_connections_currently_open INT 0
PROCESS proxy.process.nntp.server_bytes_read INT 0
PROCESS proxy.process.nntp.server_article_bytes_read INT 0
PROCESS proxy.process.nntp.server_overview_bytes_read INT 0
PROCESS proxy.process.nntp.server_bytes_written INT 0
PROCESS proxy.process.nntp.control_cancels INT 0
PROCESS proxy.process.nntp.control_newgroups INT 0
PROCESS proxy.process.nntp.control_rmgroups INT 0

PROCESS proxy.process.nntp.client_connections INT 0
PROCESS proxy.process.nntp.client_commands INT 0
PROCESS proxy.process.nntp.server_connections INT 0
PROCESS proxy.process.nntp.server_commands INT 0
PROCESS proxy.process.nntp.article_hits INT 0
PROCESS proxy.process.nntp.article_misses INT 0
PROCESS proxy.process.nntp.overview_hits INT 0
PROCESS proxy.process.nntp.overview_refreshes INT 0
PROCESS proxy.process.nntp.group_hits INT 0
PROCESS proxy.process.nntp.group_refreshes INT 0
PROCESS proxy.process.nntp.posts INT 0
PROCESS proxy.process.nntp.post_bytes INT 0
PROCESS proxy.process.nntp.pull_bytes INT 0
PROCESS proxy.process.nntp.feed_bytes INT 0

# Standardized Statistics
PROCESS proxy.process.nntp.incoming_requests INT 0
PROCESS proxy.process.nntp.outgoing_requests INT 0
PROCESS proxy.process.nntp.downstream.request_bytes INT 0
PROCESS proxy.process.nntp.downstream.response_bytes INT 0
PROCESS proxy.process.nntp.upstream.request_bytes INT 0
PROCESS proxy.process.nntp.upstream.response_bytes INT 0
PROCESS proxy.process.nntp.cache_hit_fresh INT 0
PROCESS proxy.process.nntp.cache_miss_cold INT 0
PROCESS proxy.process.nntp.transaction_counts.hit_fresh INT 0
PROCESS proxy.process.nntp.transaction_counts.miss_cold INT 0
PROCESS proxy.process.nntp.current_client_connections INT 0
PROCESS proxy.process.nntp.current_server_connections INT 0

##############################################################################
#
```

```
# Http Statistics
#
###############################################################################

# Standardized Statistics - transaction stats
PROCESS proxy.process.http.incoming_requests INT 0
PROCESS proxy.process.http.outgoing_requests INT 0

PROCESS proxy.process.http.cache_hit_fresh INT 0
PROCESS proxy.process.http.cache_hit_revalidated INT 0
PROCESS proxy.process.http.cache_miss_cold INT 0
PROCESS proxy.process.http.cache_miss_changed INT 0
PROCESS proxy.process.http.cache_miss_client_no_cache INT 0
PROCESS proxy.process.http.cache_miss_not_cacheable INT 0

# Standardized Statistics - dynamic stats
PROCESS proxy.process.http.current_client_connections INT 0
PROCESS proxy.process.http.current_parent_proxy_connections INT 0
PROCESS proxy.process.http.current_server_connections INT 0
PROCESS proxy.process.http.current_cache_connections INT 0
#
# The transaction stats
#
PROCESS proxy.process.http.incoming_responses INT 0
PROCESS proxy.process.http.invalid_client_requests INT 0
PROCESS proxy.process.http.get_requests INT 0
PROCESS proxy.process.http.head_requests INT 0
PROCESS proxy.process.http.trace_requests INT 0
PROCESS proxy.process.http.options_requests INT 0
PROCESS proxy.process.http.post_requests INT 0
PROCESS proxy.process.http.put_requests INT 0
PROCESS proxy.process.http.delete_requests INT 0
PROCESS proxy.process.http.connect_requests INT 0
PROCESS proxy.process.http.client_no_cache_requests INT 0
PROCESS proxy.process.http.broken_server_connections INT 0
PROCESS proxy.process.http.cache_lookups INT 0
PROCESS proxy.process.http.cache_hits_fresh INT 0
PROCESS proxy.process.http.cache_hits_expired INT 0
PROCESS proxy.process.http.cache_hits_unauthorized INT 0
PROCESS proxy.process.http.cache_misses INT 0
PROCESS proxy.process.http.cache_writes INT 0
PROCESS proxy.process.http.cache_updates INT 0
PROCESS proxy.process.http.cache_deletes INT 0
PROCESS proxy.process.http.tunnels INT 0
PROCESS proxy.process.http.icp_suggested_lookups INT 0
PROCESS proxy.process.http.client_transaction_time INT 0
PROCESS proxy.process.http.client_write_time INT 0
PROCESS proxy.process.http.server_read_time INT 0
PROCESS proxy.process.http.icp_transaction_time INT 0
PROCESS proxy.process.http.icp_raw_transaction_time INT 0
PROCESS proxy.process.http.parent_proxy_transaction_time INT 0
PROCESS proxy.process.http.parent_proxy_raw_transaction_time INT 0
PROCESS proxy.process.http.server_transaction_time INT 0
PROCESS proxy.process.http.server_raw_transaction_time INT 0
PROCESS proxy.process.http.user_agent_request_header_total_size INT 0
PROCESS proxy.process.http.user_agent_response_header_total_size INT 0
PROCESS proxy.process.http.user_agent_request_document_total_size INT 0
PROCESS proxy.process.http.user_agent_response_document_total_size INT 0
PROCESS proxy.process.http.origin_server_request_header_total_size INT 0
PROCESS proxy.process.http.origin_server_response_header_total_size INT 0
PROCESS proxy.process.http.origin_server_request_document_total_size INT 0
PROCESS proxy.process.http.origin_server_response_document_total_size INT 0
PROCESS proxy.process.http.response_document_size_100 INT 0
PROCESS proxy.process.http.response_document_size_1K INT 0
PROCESS proxy.process.http.response_document_size_3K INT 0
```

```
PROCESS proxy.process.http.response_document_size_5K INT 0
PROCESS proxy.process.http.response_document_size_10K INT 0
PROCESS proxy.process.http.response_document_size_1M INT 0
PROCESS proxy.process.http.response_document_size_inf INT 0
PROCESS proxy.process.http.request_document_size_100 INT 0
PROCESS proxy.process.http.request_document_size_1K INT 0
PROCESS proxy.process.http.request_document_size_3K INT 0
PROCESS proxy.process.http.request_document_size_5K INT 0
PROCESS proxy.process.http.request_document_size_10K INT 0
PROCESS proxy.process.http.request_document_size_1M INT 0
PROCESS proxy.process.http.request_document_size_inf INT 0
PROCESS proxy.process.http.user_agent_speed_bytes_per_sec_100 INT 0
PROCESS proxy.process.http.user_agent_speed_bytes_per_sec_1K INT 0
PROCESS proxy.process.http.user_agent_speed_bytes_per_sec_10K INT 0
PROCESS proxy.process.http.user_agent_speed_bytes_per_sec_100K INT 0
PROCESS proxy.process.http.user_agent_speed_bytes_per_sec_1M INT 0
PROCESS proxy.process.http.user_agent_speed_bytes_per_sec_10M INT 0
PROCESS proxy.process.http.user_agent_speed_bytes_per_sec_100M INT 0
PROCESS proxy.process.http.origin_server_speed_bytes_per_sec_100 INT 0
PROCESS proxy.process.http.origin_server_speed_bytes_per_sec_1K INT 0
PROCESS proxy.process.http.origin_server_speed_bytes_per_sec_10K INT 0
PROCESS proxy.process.http.origin_server_speed_bytes_per_sec_100K INT 0
PROCESS proxy.process.http.origin_server_speed_bytes_per_sec_1M INT 0
PROCESS proxy.process.http.origin_server_speed_bytes_per_sec_10M INT 0
PROCESS proxy.process.http.origin_server_speed_bytes_per_sec_100M INT 0
PROCESS proxy.process.http.total_transactions_time INT 0
PROCESS proxy.process.http.total_transactions_think_time INT 0

# Client's perspective stats - counts
PROCESS proxy.process.http.transaction_counts.hit_fresh INT 0
PROCESS proxy.process.http.transaction_counts.hit_revalidated INT 0
PROCESS proxy.process.http.transaction_counts.miss_cold INT 0
PROCESS proxy.process.http.transaction_counts.miss_changed INT 0
PROCESS proxy.process.http.transaction_counts.miss_client_no_cache INT 0
PROCESS proxy.process.http.transaction_counts.miss_not_cacheable INT 0
PROCESS proxy.process.http.transaction_counts.errors.aborts INT 0
PROCESS proxy.process.http.transaction_counts.errors.possible_aborts INT 0
PROCESS proxy.process.http.transaction_counts.errors.connect_failed INT 0
PROCESS proxy.process.http.transaction_counts.errors.pre_accept_hangups INT 0
PROCESS proxy.process.http.transaction_counts.errors.empty_hangups INT 0
PROCESS proxy.process.http.transaction_counts.errors.early_hangups INT 0
PROCESS proxy.process.http.transaction_counts.errors.other INT 0

# Client's perspective stats - times
PROCESS proxy.process.http.transaction_totaltime.hit_fresh FLOAT 0
PROCESS proxy.process.http.transaction_totaltime.hit_revalidated FLOAT 0
PROCESS proxy.process.http.transaction_totaltime.miss_cold FLOAT 0
PROCESS proxy.process.http.transaction_totaltime.miss_changed FLOAT 0
PROCESS proxy.process.http.transaction_totaltime.miss_client_no_cache FLOAT 0
PROCESS proxy.process.http.transaction_totaltime.miss_not_cacheable FLOAT 0
PROCESS proxy.process.http.transaction_totaltime.errors.aborts FLOAT 0
PROCESS proxy.process.http.transaction_totaltime.errors.possible_aborts FLOAT 0
PROCESS proxy.process.http.transaction_totaltime.errors.connect_failed FLOAT 0
PROCESS proxy.process.http.transaction_totaltime.errors.pre_accept_hangups FLOAT 0
PROCESS proxy.process.http.transaction_totaltime.errors.empty_hangups FLOAT 0
PROCESS proxy.process.http.transaction_totaltime.errors.early_hangups FLOAT 0
PROCESS proxy.process.http.transaction_totaltime.errors.other FLOAT 0

#
# Bandwidth Savings Transaction Stats
PROCESS proxy.process.http.tcp_hit_count_stat INT 0
PROCESS proxy.process.http.tcp_hit_user_agent_bytes_stat INT 0
PROCESS proxy.process.http.tcp_hit_origin_server_bytes_stat INT 0
PROCESS proxy.process.http.tcp_miss_count_stat INT 0
PROCESS proxy.process.http.tcp_miss_user_agent_bytes_stat INT 0
```

```
PROCESS proxy.process.http.tcp_miss_origin_server_bytes_stat INT 0
PROCESS proxy.process.http.tcp_expired_miss_count_stat INT 0
PROCESS proxy.process.http.tcp_expired_miss_user_agent_bytes_stat INT 0
PROCESS proxy.process.http.tcp_expired_miss_origin_server_bytes_stat INT 0
PROCESS proxy.process.http.tcp_refresh_hit_count_stat INT 0
PROCESS proxy.process.http.tcp_refresh_hit_user_agent_bytes_stat INT 0
PROCESS proxy.process.http.tcp_refresh_hit_origin_server_bytes_stat INT 0
PROCESS proxy.process.http.tcp_refresh_miss_count_stat INT 0
PROCESS proxy.process.http.tcp_refresh_miss_user_agent_bytes_stat INT 0
PROCESS proxy.process.http.tcp_refresh_miss_origin_server_bytes_stat INT 0
PROCESS proxy.process.http.tcp_client_refresh_count_stat INT 0
PROCESS proxy.process.http.tcp_client_refresh_user_agent_bytes_stat INT 0
PROCESS proxy.process.http.tcp_client_refresh_origin_server_bytes_stat INT 0
PROCESS proxy.process.http.tcp_ims_hit_count_stat INT 0
PROCESS proxy.process.http.tcp_ims_hit_user_agent_bytes_stat INT 0
PROCESS proxy.process.http.tcp_ims_hit_origin_server_bytes_stat INT 0
PROCESS proxy.process.http.tcp_ims_miss_count_stat INT 0
PROCESS proxy.process.http.tcp_ims_miss_user_agent_bytes_stat INT 0
PROCESS proxy.process.http.tcp_ims_miss_origin_server_bytes_stat INT 0
PROCESS proxy.process.http.err_client_abort_count_stat INT 0
PROCESS proxy.process.http.err_client_abort_user_agent_bytes_stat INT 0
PROCESS proxy.process.http.err_client_abort_origin_server_bytes_stat INT 0
PROCESS proxy.process.http.err_connect_fail_count_stat INT 0
PROCESS proxy.process.http.err_connect_fail_user_agent_bytes_stat INT 0
PROCESS proxy.process.http.err_connect_fail_origin_server_bytes_stat INT 0
PROCESS proxy.process.http.misc_count_stat INT 0
PROCESS proxy.process.http.misc_user_agent_bytes_stat INT 0
PROCESS proxy.process.http.misc_origin_server_bytes_stat INT 0
PROCESS proxy.process.http.background_fill_bytes_aborted_stat INT 0
PROCESS proxy.process.http.background_fill_bytes_completed_stat INT 0
#
# The dynamic stats
#
PROCESS proxy.process.http.background_fill_current_count INT 0
PROCESS proxy.process.http.current_client_transactions INT 0
PROCESS proxy.process.http.current_parent_proxy_transactions INT 0
PROCESS proxy.process.http.current_icp_transactions INT 0
PROCESS proxy.process.http.current_server_transactions INT 0
PROCESS proxy.process.http.current_parent_proxy_raw_transactions INT 0
PROCESS proxy.process.http.current_icp_raw_transactions INT 0
PROCESS proxy.process.http.current_server_raw_transactions INT 0
PROCESS proxy.process.http.client_connection_time INT 0
PROCESS proxy.process.http.parent_proxy_connection_time INT 0
PROCESS proxy.process.http.server_connection_time INT 0
PROCESS proxy.process.http.cache_connection_time INT 0
PROCESS proxy.process.http.avg_transactions_per_client_connection FLOAT 0
PROCESS proxy.process.http.avg_transactions_per_server_connection FLOAT 0
PROCESS proxy.process.http.avg_transactions_per_parent_connection FLOAT 0
##############################################################################
#
# SOCKS Processor
#
##############################################################################
CONFIG proxy.config.socks.socks_needed INT 0
CONFIG proxy.config.socks.socks_version FLOAT 4.0
CONFIG proxy.config.socks.socks_server_ip_str STRING 0.0.0.0
CONFIG proxy.config.socks.socks_server_port INT 1080
CONFIG proxy.config.socks.socks_config_file STRING socks.config
CONFIG proxy.config.socks.socks_timeout INT 100
PROCESS proxy.process.socks.connections_unsuccessful COUNTER 0
PROCESS proxy.process.socks.connections_successful COUNTER 0
PROCESS proxy.process.socks.connections_currently_open INT 0
##############################################################################
#
# FTP Processor
```

```
#
###############################################################################
      ########################################################
      # ftp_mode 1 PASV then PORT, 2 PORT only 3 PASV only #
      ########################################################
CONFIG proxy.config.ftp.data_connection_mode INT 1
CONFIG proxy.config.ftp.control_connection_timeout INT 300
#
# Transaction-based ftp stats
PROCESS proxy.process.ftp.cache_lookups INT 0
PROCESS proxy.process.ftp.cache_hits INT 0
PROCESS proxy.process.ftp.cache_misses INT 0
#
# Dynamic ftp stats
PROCESS proxy.process.ftp.connections_successful_pasv INT 0
PROCESS proxy.process.ftp.connections_failed_pasv INT 0
PROCESS proxy.process.ftp.connections_successful_port INT 0
PROCESS proxy.process.ftp.connections_failed_port INT 0
PROCESS proxy.process.ftp.connections_currently_open INT 0
###############################################################################
#
# Transform Processor
#
###############################################################################
PROCESS proxy.process.transform.jg_opened INT 0
PROCESS proxy.process.transform.jg_notcompressed_nocontentlength INT 0
PROCESS proxy.process.transform.jg_notcompressed_largerthanbufsize INT 0
PROCESS proxy.process.transform.jg_notcompressed_smallerthanminsize INT 0
PROCESS proxy.process.transform.jg_notcompressed_largerthanmaxsize INT 0
PROCESS proxy.process.transform.jg_notcompressed_isagif INT 0
PROCESS proxy.process.transform.jg_notcompressed_compdopenfailed INT 0
PROCESS proxy.process.transform.jg_notcompressed_nonzerostatus INT 0
PROCESS proxy.process.transform.jg_notcompressed_wrongcompressedlength INT 0
PROCESS proxy.process.transform.jg_notcompressed_unknowntunnelevent INT 0
PROCESS proxy.process.transform.jg_compression_started INT 0
PROCESS proxy.process.transform.jg_read_complete_sent_uncompressed INT 0
PROCESS proxy.process.transform.jg_read_complete_sent_compressed INT 0
PROCESS proxy.process.transform.jg_closed INT 0
PROCESS proxy.process.transform.jg_larger_than_one_buffer INT 0
PROCESS proxy.process.transform.jg_iobuffers_copied INT 0
PROCESS proxy.process.transform.jg_connects_attempted INT 0
PROCESS proxy.process.transform.jg_connects_responded INT 0
PROCESS proxy.process.transform.jg_wrongcontentlength INT 0
PROCESS proxy.process.transform.jg_writes_to_compd_finished INT 0
PROCESS proxy.process.transform.jg_reads_from_compd_finished INT 0
PROCESS proxy.process.transform.jg_msecs_start_to_connect_attempt INT 0
PROCESS proxy.process.transform.jg_msecs_start_to_connect_response INT 0
PROCESS proxy.process.transform.jg_msecs_start_to_write_to_compd_finished INT 0
PROCESS proxy.process.transform.jg_msecs_start_to_read_from_compd_finished INT 0
PROCESS proxy.process.transform.jg_compressed_input_totalsize INT 0
PROCESS proxy.process.transform.jg_compressed_output_totalsize INT 0
PROCESS proxy.process.transform.jg_notcompressed_nonzerostatus_totalsize INT 0
PROCESS proxy.process.transform.jg_notcompressed_largerthanbufsize_totalsize INT 0
PROCESS proxy.process.transform.jg_notcompressed_isagif_totalsize INT 0
PROCESS proxy.process.transform.jg_notcompressed_smallerthanminsize_totalsize INT 0
PROCESS proxy.process.transform.jg_notcompressed_largerthanmaxsize_totalsize INT 0
###############################################################################
#
# I/O Subsystem
#
###############################################################################
CONFIG proxy.config.io.max_buffer_size INT 32768
###############################################################################
#
# Event Subsystem
```

```
#
# All these configuration variables are "Engineering" variables.  They
# shouldn't be exposed to the user. (Except for max_threads)
#
##############################################################################
CONFIG proxy.config.event-subsystem.max_threads INT 1
CONFIG proxy.config.event-subsystem.total_number_of_threads INT 1
CONFIG proxy.config.event-subsystem.number_of_spawn_threads INT 1
CONFIG proxy.config.event-subsystem.number_of_call_threads INT 1
CONFIG proxy.config.event-subsystem.number_of_net_threads INT 1
CONFIG proxy.config.event-subsystem.number_of_disk_threads INT 1
CONFIG proxy.config.event-subsystem.number_of_cluster_threads INT 1
CONFIG proxy.config.event-subsystem.number_of_ftp_threads INT 1
CONFIG proxy.config.event-subsystem.number_of_cache_threads INT 1
##############################################################################
#
# Disk Subsystem
#
##############################################################################

##############################################################################
#
# Net Subsystem
#
##############################################################################
CONFIG proxy.config.net.connections_throttle INT 16000
CONFIG proxy.config.net.accept_throttle INT 0
# 0 = turn off adaptation, poll continuously - may hurt performance under load
# 1 = adapt to load levels to maximize responsiveness at all load levels
CONFIG proxy.config.net.adaptation INT 1

PROCESS proxy.process.net.read_bytes INT 0
PROCESS proxy.process.net.write_bytes INT 0
PROCESS proxy.process.net.connections_currently_open INT 0
PROCESS proxy.process.net.accepts_currently_open INT 0
PROCESS proxy.process.net.calls_to_readfromnet INT 0
PROCESS proxy.process.net.calls_to_readfromnet_afterpoll INT 0
PROCESS proxy.process.net.calls_to_read INT 0
PROCESS proxy.process.net.calls_to_read_nodata INT 0
PROCESS proxy.process.net.calls_to_writetonet INT 0
PROCESS proxy.process.net.calls_to_writetonet_afterpoll INT 0
PROCESS proxy.process.net.calls_to_write INT 0
PROCESS proxy.process.net.calls_to_write_nodata INT 0

##############################################################################
#
# Cluster Subsystem
#
##############################################################################
CONFIG proxy.config.cluster.cluster_port INT 8086
CONFIG proxy.config.cluster.cluster_configuration STRING cluster.config
CONFIG proxy.config.cluster.ethernet_interface STRING ef0
PROCESS proxy.process.cluster.connections_open   INT 0
PROCESS proxy.process.cluster.connections_opened INT 0
PROCESS proxy.process.cluster.connections_closed INT 0
PROCESS proxy.process.cluster.connections_avg_time FLOAT 0.00
PROCESS proxy.process.cluster.connections_locked INT 0
PROCESS proxy.process.cluster.op_delayed_for_lock INT 0
PROCESS proxy.process.cluster.read_bytes INT 0
PROCESS proxy.process.cluster.write_bytes INT 0
PROCESS proxy.process.cluster.control_messages_sent INT 0
PROCESS proxy.process.cluster.control_messages_received INT 0
PROCESS proxy.process.cluster.control_messages_avg_send_time FLOAT 0.0
PROCESS proxy.process.cluster.control_messages_avg_receive_time FLOAT 0.0
PROCESS proxy.process.cluster.nodes INT 0
```

```
PROCESS proxy.process.cluster.machines_allocated INT 0
PROCESS proxy.process.cluster.machines_freed INT 0
PROCESS proxy.process.cluster.configuration_changes INT 0
PROCESS proxy.process.cluster.net_backup INT 0
PROCESS proxy.process.cluster.connections_bumped INT 0
PROCESS proxy.process.cluster.delayed_reads INT 0
PROCESS proxy.process.cluster.byte_bank_used INT 0
PROCESS proxy.process.cluster.alloc_data_news INT 0
PROCESS proxy.process.cluster.write_bb_mallocs INT 0
PROCESS proxy.process.cluster.partial_reads INT 0
PROCESS proxy.process.cluster.partial_writes INT 0
#
PROCESS proxy.process.cluster.cache_outstanding INT 0
PROCESS proxy.process.cluster.remote_op_timeouts INT 0
PROCESS proxy.process.cluster.remote_op_reply_timeouts INT 0
PROCESS proxy.process.cluster.chan_inuse INT 0

# Cluster/Cache stats
#
###############################################################################
#
# Cache
#
###############################################################################
      ########################
      # cache configuration #
      ########################
CONFIG proxy.config.cache.storage_filename STRING storage.config
CONFIG proxy.config.cache.control.filename STRING cache.config
CONFIG proxy.config.cache.ip_allow.filename STRING ip_allow.config

CONFIG proxy.config.cache.avg_frag_size INT 8192
CONFIG proxy.config.cache.vary_on_user_agent INT 0
CONFIG proxy.config.cache.tie_vector_and_frag INT 1

# 0 - eventProcessor.schedule_imm
# 1 - original thread per pool mapping
# 2 - new thread per pool mapping
# 3 - round robin amongst cache threads
CONFIG proxy.config.cache.scheduling_method INT 3

# 0 - MD5 fingerprinting
# 1 - No fingerprinting
CONFIG proxy.config.cache.fingerprint_method INT 0
CONFIG proxy.config.cache.max_buffers_per_pool INT 1

      ####################
      # limits on cache #
      ####################
# UNIMPLEMENTED: CONFIG proxy.config.cache.limits.http.quota FLOAT 1.0

  # The maximum number of alternates that are allowed for any given URL.
  # It is not possible to strictly enforce this if the variable
  #   'proxy.config.cache.vary_on_user_agent' is set to 1.
  # (0 disables the maximum number of alts check)
CONFIG proxy.config.cache.limits.http.max_alts INT 3

  # The maximum size of an http document that will be stored in the cache.
  # (0 disables the maximum document size check)
CONFIG proxy.config.cache.limits.http.max_doc_size INT 0

# UNIMPLEMENTED: CONFIG proxy.config.cache.limits.nntp.quota FLOAT 1.0

  # The maximum size of an nntp document that will be stored in the cache.
  # (0 disables the maximum document size check)
```

```
CONFIG proxy.config.cache.limits.nntp.max_doc_size INT 0

# UNIMPLEMENTED: CONFIG proxy.config.cache.limits.rtsp.quota FLOAT 1.0

  # The maximum size of an rtsp document that will be stored in the cache.
  # (0 disables the maximum document size check)
CONFIG proxy.config.cache.limits.rtsp.max_doc_size INT 0

  # when should GC kick in and start cleaning
CONFIG proxy.config.cache.gc.watermark FLOAT 0.90

  # how often should the GC run
CONFIG proxy.config.cache.gc.frequency INT 10

  # how often should the directory be synced
CONFIG proxy.config.cache.dir.sync_frequency INT 60
CONFIG proxy.config.cache.dir.read_at_startup INT 1

  # how often should the directory be checked
  # (0 disables consistency checks of the directory)
CONFIG proxy.config.cache.dir.check_frequency INT 0

  # how often should the pool headers be synced (seconds)
CONFIG proxy.config.cache.pool.sync_frequency INT 30

  # how often should the write aggregation buffers should be
  # checked for flushing (milliseconds)
  # (0 means flush as quickly as possible)
CONFIG proxy.config.cache.pool.flush_frequency INT 0

  # how often should we force the flushing of the write aggregation
  # buffers to be flushed. if we don't force them to be flushed then
  # they will only be flushed when they become full.
  # NOTE: this has a tendency to waste space and is really only
  #       intended and useful for testing
  # (0 disables forced flushing of the write aggregation buffers)
CONFIG proxy.config.cache.pool.force_flush_frequency INT 0

  # how often should the hostdb be synced
CONFIG proxy.config.cache.hostdb.sync_frequency INT 60

  # default the ram cache size to 20 MB
CONFIG proxy.config.cache.ram_cache.size INT 20971520

  # how often should the RAM cache be cleaned
CONFIG proxy.config.cache.ram_cache.clean_frequency INT 60

  # internal configuration variables that control generation of
  # synthetic error conditions. THESE VALUES SHOULD NOT BE CHANGED.
CONFIG proxy.config.cache.synthetic_errors.locks FLOAT 0.0
CONFIG proxy.config.cache.synthetic_errors.disk.lseek FLOAT 0.0
CONFIG proxy.config.cache.synthetic_errors.disk.read FLOAT 0.0
CONFIG proxy.config.cache.synthetic_errors.disk.write FLOAT 0.0
CONFIG proxy.config.cache.synthetic_errors.disk.readv FLOAT 0.0
CONFIG proxy.config.cache.synthetic_errors.disk.writev FLOAT 0.0
CONFIG proxy.config.cache.synthetic_errors.disk.pread FLOAT 0.0
CONFIG proxy.config.cache.synthetic_errors.disk.pwrite FLOAT 0.0

  # cache stats

  # percent_full = bytes_used / bytes_total
PROCESS proxy.process.cache.percent_full FLOAT 0.0
PROCESS proxy.process.cache.bytes_used INT 0
PROCESS proxy.process.cache.bytes_total INT 0
```

```
  # RAM cache hit-rate
  # hit-rate = cache.ram_cache.hits / cache.ram_cache.accesses
PROCESS proxy.process.cache.ram_cache.hits INT 0
PROCESS proxy.process.cache.ram_cache.accesses INT 0
PROCESS proxy.process.cache.ram_cache.bytes.locked INT 0
PROCESS proxy.process.cache.ram_cache.bytes.unlocked INT 0

  # the instantaneous count of the number of fragments stored in the cache
  # there are usually two fragments per http object, one per nntp article
PROCESS proxy.process.cache.frag_count INT 0

  # stats for the lookup() operation [lookup() is only used by ICP]
  # avg success time = cache.lookup.success.time / cache.lookup.success
  # avg failure time = cache.lookup.failure.time / cache.lookup.failure
  # avg operation time = (cache.lookup.success.time + cache.lookup.failure.time) /
  #                      (cache.lookup.success + cache.lookup.failure)
PROCESS proxy.process.cache.lookup.active INT 0
PROCESS proxy.process.cache.lookup.success INT 0
PROCESS proxy.process.cache.lookup.success.time INT 0
PROCESS proxy.process.cache.lookup.failure INT 0
PROCESS proxy.process.cache.lookup.failure.time INT 0

  # stats for the open-read() operation
  # avg success time = cache.read.success.time / cache.read.success
  # avg failure time = cache.read.failure.time / cache.read.failure
  # avg operation time = (cache.read.success.time + cache.read.failure.time) /
  #                      (cache.read.success + cache.read.failure)
  # cache hit-rate = cache.read.success / (cache.read.success + cache.read.failure)
PROCESS proxy.process.cache.read.active INT 0
PROCESS proxy.process.cache.read.success INT 0
PROCESS proxy.process.cache.read.success.time INT 0
PROCESS proxy.process.cache.read.abort INT 0
PROCESS proxy.process.cache.read.abort.time INT 0
PROCESS proxy.process.cache.read.cancel INT 0
PROCESS proxy.process.cache.read.cancel.time INT 0
PROCESS proxy.process.cache.read.bytes INT 0
PROCESS proxy.process.cache.read.hit INT 0
PROCESS proxy.process.cache.read.miss INT 0

  # stats for the open-write() operation
  # avg success time = cache.write.success.time / cache.write.success
  # avg failure time = cache.write.failure.time / cache.write.failure
  # avg operation time = (cache.write.success.time + cache.write.failure.time) /
  # (cache.write.success + cache.write.failure)
  # usually, the only time writes fail is when the cache is full and GC
  # is not keeping up
PROCESS proxy.process.cache.write.active INT 0
PROCESS proxy.process.cache.write.success INT 0
PROCESS proxy.process.cache.write.success.time INT 0
PROCESS proxy.process.cache.write.abort INT 0
PROCESS proxy.process.cache.write.abort.time INT 0
PROCESS proxy.process.cache.write.cancel INT 0
PROCESS proxy.process.cache.write.cancel.time INT 0
PROCESS proxy.process.cache.write.bytes INT 0

  # stats for the update() operation
  # avg success time = cache.update.success.time / cache.update.success
  # avg failure time = cache.update.failure.time / cache.update.failure
  # avg operation time = (cache.update.success.time + cache.update.failure.time) /
  #                      (cache.update.success + cache.update.failure)
PROCESS proxy.process.cache.update.active INT 0
PROCESS proxy.process.cache.update.success INT 0
PROCESS proxy.process.cache.update.success.time INT 0
PROCESS proxy.process.cache.update.failure INT 0
PROCESS proxy.process.cache.update.failure.time INT 0
```

```
  # stats for the link() operation [used by NNTP]
  # avg success time = cache.link.success.time / cache.link.success
  # avg failure time = cache.link.failure.time / cache.link.failure
  # avg operation time = (cache.link.success.time + cache.link.failure.time) /
  #                      (cache.link.success + cache.link.failure)
PROCESS proxy.process.cache.link.active INT 0
PROCESS proxy.process.cache.link.success INT 0
PROCESS proxy.process.cache.link.success.time INT 0
PROCESS proxy.process.cache.link.failure INT 0
PROCESS proxy.process.cache.link.failure.time INT 0

  # stats for the deref() operation [used by NNTP]
  # avg success time = cache.deref.success.time / cache.deref.success
  # avg failure time = cache.deref.failure.time / cache.deref.failure
  # avg operation time = (cache.deref.success.time + cache.deref.failure.time) /
  #                      (cache.deref.success + cache.deref.failure)
PROCESS proxy.process.cache.deref.active INT 0
PROCESS proxy.process.cache.deref.success INT 0
PROCESS proxy.process.cache.deref.success.time INT 0
PROCESS proxy.process.cache.deref.failure INT 0
PROCESS proxy.process.cache.deref.failure.time INT 0

  # stats for the remove() operation [unused?]
  # avg success time = cache.remove.success.time / cache.remove.success
  # avg failure time = cache.remove.failure.time / cache.remove.failure
  # avg operation time = (cache.remove.success.time + cache.remove.failure.time) /
  #                      (cache.remove.success + cache.remove.failure)
PROCESS proxy.process.cache.remove.active INT 0
PROCESS proxy.process.cache.remove.success INT 0
PROCESS proxy.process.cache.remove.success.time INT 0
PROCESS proxy.process.cache.remove.failure INT 0
PROCESS proxy.process.cache.remove.failure.time INT 0

PROCESS proxy.process.cache.gc.initiated INT 0
PROCESS proxy.process.cache.gc.frags.deleted INT 0
PROCESS proxy.process.cache.gc.frags.evacuated INT 0
PROCESS proxy.process.cache.gc.vectors.deleted INT 0
PROCESS proxy.process.cache.gc.vectors.evacuated INT 0
PROCESS proxy.process.cache.gc.segments.cleaned INT 0
PROCESS proxy.process.cache.gc.segments.cleared INT 0

PROCESS proxy.process.cache.dir.syncs INT 0

PROCESS proxy.process.cache.frags_per_doc.1 INT 0
PROCESS proxy.process.cache.frags_per_doc.2 INT 0
PROCESS proxy.process.cache.frags_per_doc.3+ INT 0

PROCESS proxy.process.cache.aliased_bytes INT 0

##############################################################################
#
# DNS
#
##############################################################################
CONFIG proxy.config.dns.lookup_timeout INT 15
CONFIG proxy.config.dns.retries INT 3
CONFIG proxy.config.dns.search_default_domains INT 1
CONFIG proxy.config.dns.failover_number INT 5
CONFIG proxy.config.dns.failover_period INT 30
PROCESS proxy.process.dns.total_dns_lookups INT 0
PROCESS proxy.process.dns.lookup_avg_time FLOAT 0.00
PROCESS proxy.process.dns.success_avg_time FLOAT 0.00
PROCESS proxy.process.dns.lookup_successes INT 0
PROCESS proxy.process.dns.lookup_fails INT 0
```

```
PROCESS proxy.process.dns.retries INT 0
PROCESS proxy.process.dns.max_retries_exceeded INT 0

##############################################################################
#
# HostDB
#
##############################################################################
CONFIG proxy.config.hostdb INT 1
       # up to 511 characters, may not be changed while running
CONFIG proxy.config.hostdb.filename STRING host.db
       # in entries, may not be changed while running
CONFIG proxy.config.hostdb.size INT 200000
CONFIG proxy.config.hostdb.storage_path STRING config/internal
CONFIG proxy.config.hostdb.storage_size INT 33554432
       # in minutes (all three)
#  0 = obey, 1 = ignore, 2 = min(X,ttl), 3 = max(X,ttl)
CONFIG proxy.config.hostdb.ttl_mode INT 0
CONFIG proxy.config.hostdb.lookup_timeout INT 20
CONFIG proxy.config.hostdb.timeout INT 1440
CONFIG proxy.config.hostdb.verify_after INT 720
CONFIG proxy.config.hostdb.fail.timeout INT 0
CONFIG proxy.config.hostdb.re_dns_on_reload INT 0
       # move entries to the owner on a lookup?
CONFIG proxy.config.hostdb.migrate_on_demand INT 0
       # find DNS results on another node in the cluster?
CONFIG proxy.config.hostdb.cluster INT 1
       # find DNS results for round-robin hosts on another node in the cluster?
CONFIG proxy.config.hostdb.cluster.round_robin INT 0

PROCESS proxy.process.hostdb.total_entries INT 0
PROCESS proxy.process.hostdb.total_lookups INT 0
PROCESS proxy.process.hostdb.total_hits INT 0
PROCESS proxy.process.hostdb.ttl FLOAT 0
PROCESS proxy.process.hostdb.ttl_expires INT 0
PROCESS proxy.process.hostdb.re_dns_on_reload INT 0
PROCESS proxy.process.hostdb.bytes INT 0




##############################################################################
#
# Operating System Calls
#
##############################################################################

##############################################################################
#
# HTTP
#
##############################################################################
       ###############
       # connections #
       ###############
       #################
       # transactions #
       #################

       ###################
       # connection time #
       ###################

       ####################
       # transaction time #
```

```
       ####################
          ######## in seconds ############
          ######## in milli-seconds ############
       ###############
       # cache hits #
       ###############
       ##################
       # document size #
       ##################
       ################
       # header size #
       ################
       ####################
       # connection speed #
       ####################
       #######
       # SSL #
       #######
CONFIG proxy.config.http.ssl_ports STRING 443 563
       #########
       # Stats #
       #########
# frequency is in seconds
CONFIG proxy.config.stats.snap_frequency INT 60
        ############
        # Parsing #
        ############
##############################################################################
#
# New Logging Config
#
##############################################################################
CONFIG proxy.config.log2.logging_enabled INT 1
CONFIG proxy.config.log2.log_buffer_size INT 10240
CONFIG proxy.config.log2.max_entries_per_buffer INT 25
CONFIG proxy.config.log2.max_secs_per_buffer INT 5
CONFIG proxy.config.log2.max_space_mb_for_logs INT 100
CONFIG proxy.config.log2.max_space_mb_for_orphan_logs INT 25
CONFIG proxy.config.log2.max_space_mb_headroom INT 10
CONFIG proxy.config.log2.logging_failopen INT 0
CONFIG proxy.config.log2.proxy_on_failure INT 0
CONFIG proxy.config.log2.logfile_dir STRING /usr/people/inktomi/logs
CONFIG proxy.config.log2.config_file STRING logs.config
#
CONFIG proxy.config.log2.squid_log_enabled INT 1
CONFIG proxy.config.log2.squid_log_is_ascii INT 1
CONFIG proxy.config.log2.squid_log_name STRING squid
CONFIG proxy.config.log2.squid_log_header STRING none
#
CONFIG proxy.config.log2.common_log_enabled INT 0
CONFIG proxy.config.log2.common_log_is_ascii INT 1
CONFIG proxy.config.log2.common_log_name STRING common
CONFIG proxy.config.log2.common_log_header STRING none
#
CONFIG proxy.config.log2.extended_log_enabled INT 0
CONFIG proxy.config.log2.extended_log_is_ascii INT 1
CONFIG proxy.config.log2.extended_log_name STRING extended
CONFIG proxy.config.log2.extended_log_header STRING none
#
CONFIG proxy.config.log2.extended2_log_enabled INT 0
CONFIG proxy.config.log2.extended2_log_is_ascii INT 1
CONFIG proxy.config.log2.extended2_log_name STRING extended2
CONFIG proxy.config.log2.extended2_log_header STRING none
#
CONFIG proxy.config.log2.separate_icp_logs INT 0
```

```
CONFIG proxy.config.log2.separate_nntp_logs INT 1
CONFIG proxy.config.log2.separate_rni_logs INT 1
CONFIG proxy.config.log2.separate_host_logs INT 0
#
CONFIG proxy.config.log2.custom_logs_enabled INT 0
#
CONFIG proxy.config.log2.collation_enabled INT 0
CONFIG proxy.config.log2.collation_host STRING NULL
CONFIG proxy.config.log2.collation_port INT 8085
CONFIG proxy.config.log2.collation_secret STRING foobar
#
CONFIG proxy.config.log2.rolling_enabled INT 1
CONFIG proxy.config.log2.rolling_interval_sec INT 86400
CONFIG proxy.config.log2.rolling_offset_hr INT 0
CONFIG proxy.config.log2.auto_delete_rolled_files INT 1
#
CONFIG proxy.config.log2.sampling_frequency INT 1
CONFIG proxy.config.log2.space_used_frequency INT 1
CONFIG proxy.config.log2.file_stat_frequency INT 32
##############################################################################
#
# New Logging Stats
#
##############################################################################
#
# bytes moved
#
PROCESS proxy.process.log2.bytes_buffered INT 0
PROCESS proxy.process.log2.bytes_written_to_disk INT 0
PROCESS proxy.process.log2.bytes_sent_to_network INT 0
PROCESS proxy.process.log2.bytes_received_from_network INT 0
#
# I/O
#
PROCESS proxy.process.log2.log_files_open COUNTER 0
PROCESS proxy.process.log2.log_files_space_used INT 0
#
# events, should be COUNTERs
#
PROCESS proxy.process.log2.event_log_error COUNTER 0
PROCESS proxy.process.log2.event_log_access COUNTER 0
PROCESS proxy.process.log2.event_log_access_fail COUNTER 0
PROCESS proxy.process.log2.event_log_access_skip COUNTER 0
#
##############################################################################
#
# RNI Config
#
##############################################################################
CONFIG proxy.config.rni.enabled INT 1
CONFIG proxy.config.rni.client_port INT 8090
##############################################################################
#
# RNI Stats
#
##############################################################################
# Basic stats
#
PROCESS proxy.process.rni.object_count COUNTER 0
PROCESS proxy.process.rni.block_hit_count COUNTER 0
PROCESS proxy.process.rni.block_miss_count COUNTER 0
PROCESS proxy.process.rni.byte_hit_sum INT 0
PROCESS proxy.process.rni.byte_miss_sum INT 0
PROCESS proxy.process.rni.time_hit_sum INT 0
PROCESS proxy.process.rni.time_miss_sum INT 0
```

```
#
# Cross-protocol stats for the dashboard/node pages
#
# Downstream = to/from user agent
# Upstream = to/from origin server
#
PROCESS proxy.process.rni.downstream_requests INT 0
PROCESS proxy.process.rni.downstream.request_bytes  INT 0
PROCESS proxy.process.rni.downstream.response_bytes INT 0
#
PROCESS proxy.process.rni.upstream_requests INT 0
PROCESS proxy.process.rni.upstream.request_bytes  INT 0
PROCESS proxy.process.rni.upstream.response_bytes INT 0
#
PROCESS proxy.process.rni.current_client_connections INT 0
PROCESS proxy.process.rni.current_server_connections INT 0
PROCESS proxy.process.rni.current_cache_connections INT 0
#
PROCESS proxy.process.rni.errors.aborts INT 0
PROCESS proxy.process.rni.errors.connect_failed INT 0
PROCESS proxy.process.rni.errors.other INT 0
#
# Derivable RNI stats
#
# total_blocks_served = block_hit_count + block_miss_count
# total_bytes_served = byte_hit_sum + byte_miss_sum
# total_time_spent = time_hit_sum + time_miss_sum
# ave_blocks_per_object = total_blocks_served / object_count
# ave_hit_time = time_hit_sum / block_hit_count
# ave_miss_time = time_miss_sum / block_miss_count
# bandwidth_savings = time_hit_sum / total_time_spent
# ...
##############################################################################
#
# Content Filtering
#
##############################################################################
CONFIG proxy.config.content_filter.filename STRING filter.config

##############################################################################
#
# Reverse Proxy
#
##############################################################################
CONFIG proxy.config.reverse_proxy.enabled INT 0
CONFIG proxy.config.url_remap.default_to_server_pac INT 0
CONFIG proxy.config.url_remap.filename STRING remap.config
CONFIG proxy.config.url_remap.remap_required INT 0
#
##############################################################################
# ICP Configuration
##############################################################################
#      enabled=0 ICP disabled
#      enabled=1 Allow receive of ICP queries
#      enabled=2 Allow send/receive of ICP queries
##############################################################################
CONFIG proxy.config.icp.enabled INT 0
#
CONFIG proxy.config.icp.icp_interface STRING ef0
CONFIG proxy.config.icp.icp_port INT 3130
CONFIG proxy.config.icp.multicast_enabled INT 0
CONFIG proxy.config.icp.query_timeout INT 2
CONFIG proxy.config.icp.icp_configuration STRING icp.config
CONFIG proxy.config.icp.lookup_local INT 0
```

```
PROCESS proxy.process.icp.config_mgmt_callouts INT 0
PROCESS proxy.process.icp.reconfig_polls INT 0
PROCESS proxy.process.icp.reconfig_events INT 0
PROCESS proxy.process.icp.invalid_poll_data INT 0
PROCESS proxy.process.icp.icp_incoming_nolock INT 0
PROCESS proxy.process.icp.no_data_read INT 0
PROCESS proxy.process.icp.short_read INT 0
PROCESS proxy.process.icp.invalid_sender INT 0
PROCESS proxy.process.icp.read_not_v2_icp INT 0
PROCESS proxy.process.icp.icp_remote_query_requests INT 0
PROCESS proxy.process.icp.icp_remote_responses INT 0
PROCESS proxy.process.icp.cache_lookup_success INT 0
PROCESS proxy.process.icp.cache_lookup_fail INT 0
PROCESS proxy.process.icp.query_response_write INT 0
PROCESS proxy.process.icp.query_response_partial_write INT 0
PROCESS proxy.process.icp.no_icp_request_for_response INT 0
PROCESS proxy.process.icp.icp_response_request_nolock INT 0
PROCESS proxy.process.icp.icp_response_not_active_nolock INT 0
PROCESS proxy.process.icp.icp_start_icpoff INT 0
PROCESS proxy.process.icp.icp_start_nolock INT 0
PROCESS proxy.process.icp.send_query_partial_write INT 0
PROCESS proxy.process.icp.icp_queries_no_expected_replies INT 0
PROCESS proxy.process.icp.icp_not_active_nolock INT 0
PROCESS proxy.process.icp.icp_query_hits INT 0
PROCESS proxy.process.icp.icp_query_misses INT 0
PROCESS proxy.process.icp.invalid_icp_query_response INT 0
PROCESS proxy.process.icp.icp_query_requests INT 0
PROCESS proxy.process.icp.total_icp_response_time FLOAT 0.00
PROCESS proxy.process.icp.total_udp_send_queries INT 0
PROCESS proxy.process.icp.total_icp_request_time FLOAT 0.00

##############################################################################
# SNMP Configuration
##############################################################################
# if subagent_enabled is 0, but master_agent_enabled, then snmpd will
# run, but traffic manager won't be registered with it.  You'll want
# master_agent_enabled if you want to monitor the non traffic server
# MIBs: ip statistics, host stats, etc.
CONFIG proxy.config.snmp.master_agent_enabled INT 1
CONFIG proxy.config.snmp.subagent_enabled INT 1
CONFIG proxy.config.snmp.snmp_startstop STRING /etc/rc3.d/S25snmpd

# WCCP Configuration
CONFIG proxy.config.enigma.enabled INT 0
CONFIG proxy.config.enigma.router_ip STRING NULL
CONFIG proxy.config.enigma.my_ip STRING NULL
```

remap.config - URL remapping configuration file (used in reverse proxy situations and configured through the rewrite rules in the Traffic Manager UI server configuration page). When the Traffic Server acts as a server accelerator (reverse proxy) for a particular origin server, URL requests to the origin server must be mapped to the appropriate location on the Traffic Server. This file also contains rules for modifying location headers. Be sure to re-read the configuration file with traffic_line -x.

```
#$Id: remap.config,v 1.2 1998/01/14 21:04:33 clarsen Exp $
#
#   URL Remapping Config File
#
#  Format is:
#   <host>:<port>        <host>:<port>
#
#   With first one is the "From" host and the second is the "To"
#    host.  Both port numbers are optional.
```

```
#
#    If there is not a "From" port, then all access to the "From"
#    host will be remapped.  If there is not a "To" port than,
#    only the hostname will be remapped and the "To" port will
#    be the same as the port specified in the request
#
#    Examples:
#    map http://www.inktomi.com/ http://server1.real-ink.com/x/
#    remap:  http://server1.real-ink.com/x http://www.inktomi.com
```

snmpd.cnf - This file contains parameters that control user access to MIB information and trap destinations. It's 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. Be sure to re-read the configuration file with traffic_line -x.

```
# Entry type: sysDescr
# Entry format: octetString
sysDescr  "SNMPv2 agent from SNMP Research, Inc."


# Entry type: sysObjectID
# Entry format: OID
sysObjectID  solarisEMANATEMasterAgent


# Entry type: sysLocation
# Entry format: octetString
sysLocation  "Down on the farm"


# Entry type: sysContact
# Entry format: octetString
sysContact  "SNMP Research, Inc.  +1 423 573 1434"


# Entry type: sysName
# Entry format: octetString
sysName  -


# Entry type: snmpEnableAuthenTraps
# Entry format: integer
snmpEnableAuthenTraps  1


# Entry type: MAX_THREADS
# Entry format: integer
MAX_THREADS  10


# Entry type: MAX_PDU_TIME
# Entry format: integer
MAX_PDU_TIME  2500


# Entry type: RETRY_INTERVAL
# Entry format: integer
RETRY_INTERVAL  5


# Entry type: MAX_OUTPUT_WAITING
# Entry format: integer
MAX_OUTPUT_WAITING  65536
```

```
# Entry type: MAX_SUBAGENTS
# Entry format: integer
MAX_SUBAGENTS  10


# Entry type: subagent
# Entry format: octetString


# Entry type: snmpBoots
# Entry format: integer
snmpBoots  29

#Entry type: userNameEntry
 #Format:  userAuthSnmpID  (octetString)
 #         userName  (text)
 #         userGroupName  (text)
 #         userTransportLabel  (text)
 #         userMemoryType  (nonVolatile, permanent, readOnly)
#userNameEntry  localSnmpID AnneXPert HelpDesk Headquarters nonVolatile
#userNameEntry  localSnmpID BobBBookkeeper Staff StaffOffices nonVolatile
#userNameEntry  localSnmpID CharlieDChief SystemAdmin - nonVolatile
#userNameEntry  localSnmpID EarlERiser DayOperator Headquarters nonVolatile
#userNameEntry  localSnmpID Guest Guest ConferenceRoom nonVolatile
#userNameEntry  localSnmpID ShawnNShipping Staff StaffOffices nonVolatile
#userNameEntry  localSnmpID TonyaTTyper Staff StaffOffices nonVolatile
#userNameEntry  localSnmpID WyleUSleep NightOperator Headquarters nonVolatile

#Entry type: v2ContextEntry
 #Format:  v2ContextSnmpID  (octetString)
 #         v2ContextName  (text)
 #         v2ContextLocalEntity  (text)
 #         v2ContextLocalTime  (integer)
 #         v2ContextMemoryType  (nonVolatile, permanent, readOnly)
#v2ContextEntry  localSnmpID UPS1 - 1 nonVolatile
#v2ContextEntry  localSnmpID UPS2 - 1 nonVolatile

#Entry type: viewTreeEntry
 #Format:  viewTreeName  (text)
 #         viewTreeSubTree  (OID)
 #         viewTreeMask  (octetString)
 #         viewTreeType  (included, excluded)
 #         viewTreeMemoryType  (nonVolatile, permanent, readOnly)
viewTreeEntry  All 0.0 - included nonVolatile
viewTreeEntry  All iso - included nonVolatile
#viewTreeEntry  Network mib_2 - included nonVolatile
#viewTreeEntry  Network snmpTrap - included nonVolatile
#viewTreeEntry  Network snmpTraps - included nonVolatile
#viewTreeEntry  DemoRead system - included nonVolatile
#viewTreeEntry  DemoRead ifEntry.0.2 ff:bf included nonVolatile
#viewTreeEntry  DemoRead srExamples - included nonVolatile
#viewTreeEntry  DemoRead snmpTrap - included nonVolatile
#viewTreeEntry  DemoRead snmpTraps - included nonVolatile
#viewTreeEntry  HtmlPage htmlpage - included nonVolatile
#viewTreeEntry  Unsecure system - included nonVolatile
#viewTreeEntry  Unsecure snmpTrap - included nonVolatile
#viewTreeEntry  Unsecure snmpTraps - excluded nonVolatile
#viewTreeEntry  DemoWrite srExamples.1 - included nonVolatile
#viewTreeEntry  Confidential system - included nonVolatile
#viewTreeEntry  Confidential enterprises - included nonVolatile
#viewTreeEntry  Confidential snmpTrap - included nonVolatile
#viewTreeEntry  Confidential snmpTraps - included nonVolatile
```

```
#Entry type: acEntry
 #Format:  acSPI  (snmpv1, snmpv2c, usecNoAuth, usecAuth, usecPriv)
 #         acGroupName  (text)
 #         acContextName  (text)
 #         acContextNameMask  (octetString)
 #         acPrivs  (nothing, readOnly, readWrite)
 #         acReadViewName  (text)
 #         acWriteViewName  (text)
 #         acMemoryType  (nonVolatile, permanent, readOnly)
acEntry  snmpv1 Anyone default - readOnly All - nonVolatile
acEntry  snmpv2c Anyone default - readOnly All - nonVolatile
#acEntry  usecNoAuth Guest default - readWrite DemoRead DemoWrite nonVolatile
#acEntry  usecNoAuth DayOperator UPS e0 readOnly All - nonVolatile
#acEntry  usecNoAuth DayOperator default - readOnly All - nonVolatile
#acEntry  usecNoAuth SystemAdmin UPS e0 readOnly All - nonVolatile
#acEntry  usecNoAuth SystemAdmin default - readOnly All - nonVolatile
#acEntry  usecNoAuth NightOperator UPS e0 readOnly All - nonVolatile
#acEntry  usecNoAuth NightOperator default - readOnly All - nonVolatile

#Entry type: communityEntry
 #Format:  communityAuthSnmpID  (octetString)
 #         communityName  (text)
 #         communityGroupName  (text)
 #         communityContextSnmpID  (octetString)
 #         communityContextName  (text)
 #         communityTransportLabel  (text)
 #         communityMemoryType  (nonVolatile, permanent, readOnly)
communityEntry  localSnmpID public Anyone localSnmpID default - nonVolatile

#Entry type: notifyEntry
 #Format:  notifyIndex  (integer)
 #         notifySPI  (snmpv1, snmpv2c, usecNoAuth, usecAuth, usecPriv)
 #         notifyIdentityName  (text)
 #         notifyTransportLabel  (text)
 #         notifyContextName  (text)
 #         notifyViewName  (text)
 #         notifyMemoryType  (nonVolatile, permanent, readOnly)
notifyEntry  1 snmpv1 public Console default All nonVolatile
notifyEntry  2 snmpv2c public Console default All nonVolatile
#notifyEntry  3 usecNoAuth CharlieDChief TrapSink default All nonVolatile
notifyEntry  4 snmpv1 public DevTest default All nonVolatile
notifyEntry  5 snmpv2c public DevTest default All nonVolatile


#Entry type: transportEntry
 #Format:  transportLabel  (text)
 #         transportSubIndex  (integer)
 #         transportDomain  (snmpUDPDomain, snmpIPXDomain, etc.)
 #         transportAddress  (transport address, i.e. 192.147.142.254:0)
 #         transportReceiveMask  (transport mask, i.e. 255.255.255.252:0)
 #         transportMMS  (integer)
 #         transportMemoryType  (nonVolatile, permanent, readOnly)
transportEntry  Console 1 snmpUDPDomain 127.0.0.1:0 255.255.255.255:0 1500 \
    nonVolatile
transportEntry  DevTest 1 snmpUDPDomain 209.1.32.44:0 255.255.255.255:0 1500 \
    nonVolatile
#transportEntry  TrapSink 1 snmpUDPDomain 192.147.142.254:0 255.255.255.255:0 \
#    1500 nonVolatile
#transportEntry  Headquarters 1 snmpUDPDomain 192.147.142.0:0 255.255.255.0:0 \
#    1500 nonVolatile
#transportEntry  Headquarters 2 snmpUDPDomain 127.0.0.1:0 255.255.255.255:0 \
#    1500 nonVolatile
#transportEntry  StaffOffices 1 snmpUDPDomain 192.147.142.0:0 255.255.255.40:0 \
#    1500 nonVolatile
#transportEntry  StaffOffices 2 snmpUDPDomain 127.0.0.1:0 255.255.255.255:0 \
```

```
#    1500 nonVolatile
#transportEntry  ConferenceRoom 1 snmpUDPDomain 192.147.142.110:0 \
#    255.255.255.255:0 1500 nonVolatile
#transportEntry  ConferenceRoom 2 snmpUDPDomain 127.0.0.1:0 255.255.255.255:0 \
#    1500 nonVolatile
```

socks.config – Used to configure the Traffic Server to work in conjunction with a Socks Server. Configured through the Traffic Manager GUI security configuration page. Be sure to re-read the configuration file with traffic_line -x.

```
#$Id: socks.config,v 1.2 1998/01/14 21:04:39 clarsen Exp $
# socks.config format:
# File consists of several lines. Each line can be a maximum of 400
# chars.
# Any line whose first string (ignoring the white spaces) is not
# equal to "no_socks" is ignored.
# Any line whose first string is "no_socks" must be of the following
# format:
# no_socks <comma separated list of IPADDRESS and/or IPADDRESS_RANGE>
# IPADDRESS is x1.x2.x3.x4
# IPADDRESS_RANGE is x1.x2.x3.x4 - y1.y2.y3.y4

no_socks   123.14.15.1 - 123.14.17.4, 123.14.15.2
no_socks   123.14.93.1 - 123.14.17.4, 123.43.15.2
no_socks   123.14.84.1 - 123.14.89.4, 123.32.15.2
```

storage.config – Lists all the files, directories and partitions that make up the Traffic Server cache. Edited through the installation process. Be sure to re-read the configuration file with traffic_line -x.

```
#
# Storage Configuration file
#
#
# The storage configuration is a list of all the storage to
# be used by the cache.
#
################################################################
##              IRIX Specific Configuration                   ##
################################################################
# Example:  using a raw disk
#
#    /dev/rdsk/dks0d2s7
#    /dev/rdsk/dks0d3s7
#
# Use partition 7 of any non-mounted option disk.
#
/dev/rdsk/dks0d2s7
```

vaddrs.config - lists all virtual IP addresses in use by Traffic Server. Not modifiable by users.

```
#
# Virtual Address Configuration file
#
# DO NOT EDIT, FILE IS MACHINE GENERATED
#
```

Traffic Server Logging

The logs directory will contain logfiles generated by the Traffic Server. Traffic Server will automatically rotate and timestamp the logfiles with each restart of the Traffic Server or at periodic intervals defined through the Traffic Manager GUI logging configuration page. Parameters can also be set for maximum logfile size and automatic deletion of oldest logs upon reaching a minimum free "headspace" threshold.

Traffic Server logs information in one or more of the following four formats. Additional information about these formats is provided in the Additional Materials section of you workbook.

1) **Squid Log** - This is equivalent to the access.log Native Format used by the Squid freeware Internet Cache program. Client accesses through Traffic Server are logged in the following format:
2) **Netscape Common Log**
3) **Netscape Extended Log**
4) **Netscape Extended-2 Log**
5) **Custom Logs** – Configured through the Traffic Manager GUI logging configuration page and defined in logs.config.(rarely used)

Starting the Traffic Server

Once you've successfully installed the software on all of the nodes in your cluster, you are ready to start the Traffic Server. Keep in mind that each node in the cluster must be started separately.

1. Log into the Traffic Server's user account
2. Set the working directory to /inktomi/bin
3. Start the traffic_cop process by typing:
   `./start_traffic_server`

This process starts up the chain of interdependent processes, which start and run the Traffic Server. Once the Traffic Server is running with start-up configuration values, you can begin to tune your server for maximum performance.

The best test of a successful installation is to point at the graphical administrator port, to review configuration and monitor the Traffic Server's activities.

Special Class Tools

Special class tools are available for populating the cache from the command line to allow you to test your Traffic Server, see the impact on cache as content increases, and to monitor and analyze logs.

This tool was developed to populate your cache by simulating usage. You are welcome to take a copy of this tool with you to your site.

```
$ ./ts_client -h
./ts_client: option requires an argument -- h
usage: ./ts_client
  -h <proxy_host>          -   no default
  -p <proxy_port>          -   no default
  -H <request_host>        -   no default (more than one -H -P combination
                               spreads load across them)
```

```
  -P <request port>         -   no default
  -n <# users>              -   defaults to 1
  -N <# loadgen clients>    -   defaults to 1 (# of users is divided by this)
  -T <run time in seconds>  -   defaults to 60
  -g <probability>          -   defaults to 0 - chance that character in request
                                will be random garbage
  -i proxy-node-ip          -   address  (one or more -i's to do cluster vs.
                                local op stats)
  -b bits/second            -   defaults to infinite (-1)
  -l <seconds>              -   defaults to 10,number of seconds of allowed
                                inactivity on individual connection after which a
                                lockup is flagged
  -v <verbose=1/0>          -   defaults to non-verbose (0)
  -s <size>                 -   fixed request size (inkBench by default)
  -r <hotset access ratio>  -   default is 0.4 (40% hit 60% miss)
  -S <hotset size>          -   default is 1000
  -q squid cache            --  no via codes
  -V 0|1                    --  Via type 0 (v1.1) or 1 (raptor)
  -a <# user agents>        --  use this many distinct user agents (default = 4)
  -u 0|1                    -- 1 == unknown Via code ok
  -U 0|1                    -- 1 == Via mismatch ok
  -R seconds                -- write out progress report each period
$ exit
```

Progress Check

After this lab, you should be able to:

1. ***Understand the steps to prepare a system for Traffic Server Installation***
2. ***Install the Traffic Server and test your installation***
3. ***Review the installation environment***

Unit 2 Practice Lab

Objectives for Unit 2 are to understand the procedures to follow to install the product and to review the environment created after installation. Once the server is installed, you will use a special tool we developed for this class, to populate the cache and allow you to begin monitoring activities.

1. Reopen your terminal window and log in as root ("su - root") (password is root). Move to the Traffic Server's bin directory (**`/space/inktomi/2.1/bin`**). Since we pre-installed the Traffic Server, you will need to remove it before you can install your new copy. The bin directory contains scripts that start and stop the Traffic Server. Execute **`./stop_traffic_server`**. You should see messages about the various processes being stopped and may get a message from Netscape that the server is no longer accepting connections.

   Move to your class_tools directory (**`/space/inktomi/class/train1/class_tools`**). Here you will find a script that will delete the current installation. Execute **`./deleteTS.sh`** and respond yes (y) when prompted to delete. Again, this may take a few moments.

2. Move to the class_software directory. Here you will find the install.sh script. For most of the install, you will be accepting defaults, but there are a few places where you will need to make changes. Using the following responses, install the Traffic Server (**`./install.sh`**).

   a) Your Traffic Server name will be your login name (for instance **`train1`**).
   b) Accept the destination directory to /space/inktomi/class/train*n*/2.1
   c) Choose yes (y) to create the directory when prompted
   d) Accept the default for the log directory (creating as needed and notice it has been updated based on your install directory)
   e) Say no to installing as part of a cluster
   f) Say no to configuring as a server accelerator (reverse proxy)
   g) Accept the default starting port
   h) Select 0 - for no changes required after the ports display
   i) Say no to installing Adaptive Redirection Module for transparency.
   j) Enter inkstudent@inktomi.com for the administrator's email address.
   k) Enter the administrator's username of "admin" and the password "admin." You will be asked to verify the password and will not see anything as you type it.
   l) Choose S to select all available disk drives for cache
   m) Choose D (done) to complete the installation (this may take a few moments, as it is the point at which the software is actually being installed).

3. Reboot the system (this will take a few minutes) and log back in, opening a new terminal window. Start the Traffic Server from the bin directory (**`./start_traffic_server`**). Use grep to make sure all three processes (traffic_server, traffic_manager and traffic_cop) have started (**`ps –ef |grep traf`**), then start your browser and click on your "Home" icon or select your Traffic Manager bookmark. Remember that your login and password for Traffic Manager are both 'admin.' Don't expect to see too much yet under the monitor mode, as there is no real activity running against the server. Return to the command line and review the installation environment and the TSinstall.log file. This file contains all the important details from your installation. Note the paths to important logging directories and keep in mind that system messages are written to /var/adm/messages.

4. Return to the class_tools directory. You will need to update your synthetic client to point to your server's IP address. Review the two files. The class_server script creates a synthetic server on your machines port 10000. The class_client script references both your Traffic Server IP and port and the machine's IP and port. When you have modified the class_client script to point at your IP address (leave ports as they are) you can execute each of the special scripts (from a new window, and in background) to begin populating the cache. Enter '**`./class_server &`**' to start a synthetic server on port 10,000 and then '**`./class_client &`**' to start moving synthetic traffic through the server.

5. Reopen the Traffic Manager in the browser and monitor activities and view graphs, paying particular attention to the Dashboard. The Dashboard will show the number of objects served. This number will slowly increase based on requests against the synthetic server. Click "Cache" to see the number of misses and "Other" to see the total lookups, hits, and information on logging space.

6. Return to the command line and to the bin directory. You will now install mrtg to enhance Traffic Server's reporting of activities.

   Move to the bin directory and start mrtg with **`./mrtgcron start`**. Return to your browser and access the mrtg page (http://{server IP}:8081/mrtg/index.html). Explore the mrtg reports and be sure to scroll down to the bottom of the page to take a look at daily and weekly lists of reports. Spend some time reviewing available options noting that it takes 5 minutes for statistics to update.

7. Set your browser to start proxying through your Traffic Server port. Go to a site of interest to you – like Yahoo.com. As you read through these documents, they will be added to your Traffic Server cache. Position your browser window to the left and your terminal window to the right. In the terminal window, go to the logs directory and view the most recent entries in the squid.log file (**`tail –f squid.log`**). Currently this is showing activity from the synthetic server.

   Modify your Netscape proxy settings (Edit, Preferences, Advanced, Proxies) choosing "Manual Proxy Configuration." Click View and enter your server IP address and Traffic Server port (8080). Now as you click around in the browser, you will see documents referenced in the squid log as they are added to the cache. You can kill the synthetic server process if the responses are getting in your way. Complete this by resetting your proxy setting to "Direct Connection" as we will work through other ways to direct traffic to the Traffic Server.

UNIT 3: CONFIGURING THE TRAFFIC SERVER

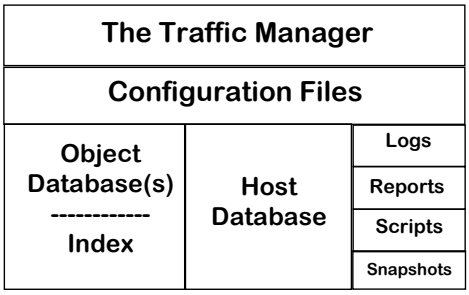
Objectives for this Unit:

- ✓ Explore Configuration Options
- ✓ Make Changes to Default Configuration Settings
- ✓ Review Impacts of Change on Traffic Server

Traffic Server Architecture

The Traffic Server consists of files, programs and processes that work together to manage and monitor caching activities. The information entered during install, most of which was to accept default values, provide the general start up configuration for your Traffic Server.

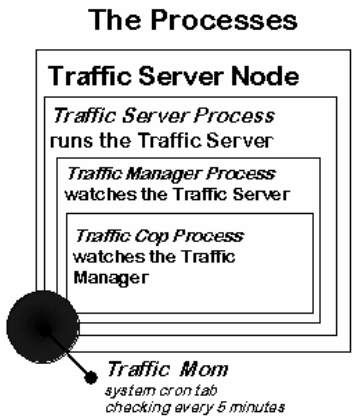
The Files



As you saw after installation of your Traffic Server, the files and programs include the Traffic Manager executables, the configuration files and various logs, reports, scripts and configuration snapshots. The databases track information about each of the cached objects and DNS information to ensure fast IP resolution.

The Processes

The robust fault-tolerant architecture is comprised of three essential and interdependent processes, working together with fault-resistant coupled clustering.



Configuring the Traffic Server

Each node must be configured separately. The following information provides configuration options for each of the components comprising the Traffic Server. From the Traffic Manager's home page, click the Configure toggle switch.

Server Basics :	**General Server Options**
Option	**Description**
On/Off	The server should generally remain ON (green light). OFF shuts down all caching and proxying services on a node-by-node basis. You must shutdown before performing some maintenance tasks.
Name	This is the DNS round-robin (the name of the cluster which includes all of the nodes).
Port	Specifies the name of the port through which users (browsers) will connect to the proxy process. The default is 8080 and this must be a dedicated port.
User ID	This is the user ID for the proxy process. This comes from the user account specified during installation (inktomi). Changes require a restart of the Traffic Manager process. Cannot be edited here.
Local Domain Expansion	Turn this on if you want unqualified host names expanded to the local domain. If user requests wolverine and Traffic Server's local domain is inktomi.com, Traffic Server will expand the host name to wolverine.inktomi.com.
.com Domain Expansion	Tells the Traffic Server to automatically preface host names with www. and suffix them with .com. If local domain expansion is on, Traffic Server will only perform .com expansion after local domain expansion has failed.

Server Basics: Web Management Options

Allows the administrator to restart the Traffic Manager and to choose how often to update displays of graphs and statistics

Option	**Description**
Restart Manager Process	Restarts the Traffic Manager, which in turn restarts the entire Traffic Server cluster. Required if changing ports and virtual IP.
Traffic Manager Port	This is the DNS round-robin (the name of the cluster which includes all of the nodes. Default is 8081.

Refresh Rate In Monitor Mode	**This dictates how often graphs and statistics will be refreshed on screen. Presents 10 settings from 10 seconds to 10 minutes (30 seconds is the default).**

Server Basics : Setting Virtual IP Addressing

The Traffic Server needs a pool of IP addresses that can be assigned to nodes as necessary, to assure that another node in the cluster can assume responsibilities should a node fail. Virtual IP addresses are first assigned during installation and can be modified. Beware that incorrect IP addressing can disable your system and that any changes to virtual IP addresses will only be picked up after restart.

Option	Description
Virtual IP On/Off	**Sets virtual IP addressing on or off. If off, Traffic Server nodes cannot cover one another's failures.**
Edit Virtual IP Addresses	**Allows you to change the list of virtual IP addresses. Changes do not take place until after restart.**

Setting Browser Auto-Configuration Options

Web browsers use the Traffic Server by specifying a preference to use a proxy server. Users set their browsers to use a proxy and provide the auto-configuration port you enter here. When the proxy port is accessed, the auto-configuration script file is downloaded. Here is an example of this file, which basically states that if the beginning of the URL is valid (http, ftp, etc.) then load the file directly from the proxy server port.

```
function FindProxyForURL(url, host) {

   // Make sure this a protocol we proxy
   if(!((url.substring(0,5) == "http:") ||
           (url.substring(0,4) == "ftp:") ||
           (url.substring(0,6) == "https:"))) {
            return "DIRECT";
   }
   return "PROXY ink-proxy.inktomi.com:8090;" +
          "PROXY proxydev.inktomi.com:8090;" +
          "DIRECT";
}
```

Option	Description
Auto-Configuration File	**Click this link to create or edit a script file containing information to automatically configure user browsers to use the Traffic Server.**

Auto-Configuration Port	**Specify the port to use for down-loading the auto-configuration file. This port cannot be assigned to any other process and changes do not become effective until you restart the server.**

Server Basics: Throttling Network Connections

You can restrict the number of network connections the Traffic Server will accept by setting a throttling limit. This limit helps to prevent system overload when traffic bottlenecks develop. Adjust the default number down if you are seeing poor performance.

Server Basics: Configuring SNMP Agents

SNMP agents can be enabled for monitoring your Traffic Server's performance. This allows you to view information about the Traffic Server and send messages (called SNMP traps) to SNMP monitoring stations. Traffic Server's SNMP agent supports access to two management information bases (MIBs): MIB-2 (a standard MIB) and the Inktomi Traffic Server MIB.

Options	Description
SNMP Master Agent	**Turms master agent on or off (for the entire cluster). This option enables access to MIB-2 information only. MIB-2 info is node specific.**
SNMP Traffic Manager MIB	**Allows access to Traffic Server MIB information. See /config/inktomi-ts-mib.my file for details on variables.**

You should configure your system so that only certain hosts can access MIBs. You configure access control and SNMP trap destinations in the /config/snmpd.cnf file.

Protocols: Configuring HTTP, NNTP, HTTPS and FTP

You can tune HTTP, NNTP& FTP timeouts and set privacy options and you can configure the Traffic Server's handling of HTTPS.

Protocols: Configuring HTTP

Options	Description
Consult ICP	**Tells the Traffic Server to send ICP queries to its ICP hierarchy in the event of a cache miss. If both ICP and HTTP parent caching are enabled, the Traffic Server first sends an ICP query. If it misses, the HTTP parent cache is checked and if it misses, the request goes to the origin server.**
Keep-alive Timeout (inbound)	**Specifies how long the Traffic Server should keep connections to user open - for a subsequent request after a transaction ends. The Traffic Server waits for this period before closing the connection. Timeout period starts over with a new request.**
Keep-alive	**Specifies how long the Traffic Server should keep connections to content**

Timeout (outbound)	**servers open for transfer of data.**
Inactivity Timeout (inbound)	**Indicates how long to keep connections to users open if a transaction stalls.**
Inactivity Timeout (outbound)	**Indicates how long to keep connections to Web servers if a transaction stalls.**
Activity Timeout (inbound)	**Specifies the maximum time the Traffic Server should remain connected to a user. If the user does not finish reading and writing data - before this timeout expires, the connection is closed.**
Activity Timeout (outbound)	**Specifies the maximum time the Traffic Server should wait for fulfillment of a request to a Web server.**
Remove Headers	**You can remove any of the following headers (which accompany transactions) to protect the privacy of your site:** **- From (identifies the user's email address)** **- Referred (identifies the Web link followed by the user)** **- User-Agent (identifies the browser making the request)** **- Cookie (identifies the user that made the request)**
Language	**Messages are (by default) displayed only in English.**

Protocols: Configuring NNTP

Options	Description
NNTP Server On/Off	**Enables Traffic Server to cache and serve news articles. If you turn NNTP on or off, you must restart the Traffic Server to affect the change.**
NNTP Server Port	**Port used for serving NNTP requests (default is 119). Keep in mind the Traffic Server's manager process must run as root to connect to port numbers less than 1024.**
Connect Message (posting allowed)	**The message that is displayed to news readers when they connect to the Traffic Server, in the posting allowed case.**
Connect Message (posting not allowed)	**The message that is displayed to news readers when they connect to the Traffic Server, in the posting not allowed case.**

NNTP Options:	
Caching	**Allows caching on NNTP articles.**
Posting	**Allows users to post NNTP articles to parent NNTP servers.**
Access Control	**Turns access control on or off (refine in the /config/nntp_access.config file).**
Authentication Server	**Runs an authentication server on this Traffic Server node. Configure which client groups must be authenticated in the /config/nntp_acces.config file.**
Clustering	**Allows cluster-wide NNTP caching.**
Allow Feeds	**Allows Traffic Server to accept feeds of new articles from feed or push groups (define these in /config/nntp_servers.config) file. The Traffic Server does not cache news articles from feed groups; instead it receives feeds of news articles as the parent NNTP server receives feeds. Push groups are groups for which the Traffic Server can both retrieve articles on demand and receive news feeds.**
Logging	**Logs NNTP transactions in access logs.**
Background Posting	**Posts NNTP articles to parent NNTP servers in background.**
Obey Cancels on Control Channel	**Sets Traffic Server to obey cancel control messages (deleting the corresponding article from the cache).**
Obey Newgroups on Control Channel	**Sets Traffic Server to obey newgroup control messages**
Obey Rmgroups on Control Channel	**Sets Traffic Server to obey rmgroup (remove group) control messages.**
Inactivity Timeout	**Sets number of seconds an idle connection remains open.**
Check for New Groups Every	**Time period between checks to poll parent NNTP servers for new groups (okay to do infrequently - parent groups lists do not change often).**
Check for Cancelled Articles Every	**Time period between checks to poll non-feed news groups.**
Check Parent NNTP Server Every	**Time period between checks to poll parent NNTP server for new articles.**
Check Cluster	**How often the Traffic Serer polls other Traffic Server nodes**

Every	**in the cluster to see if new articles have appeared.**
Check Pull Groups Every	**Traffic Server actively caches news articles, pulling them from pull groups rather than waiting for user requests. These groups are designated in /config/nntp_servers.config. How often.**
Authentication Server Host	**Host name of the authentication server.**
Authentication Server Port	**Locally run authentication server will accept connections on this port and the Traffic Server will connect to the authentication server on this port.**
Authentication Server Timeout	**How long to wait to abort an authentication operation before client has to retry.**
Client Speed Throttle	**Clients are limited to downloading no more than this number of bytes per second. A throttle of 0 (zero) means that downloading is not limited.**

Protocols: Configuring HTTPS

You can restrict SSL connections to certain ports by entering the port numbers on the Protocols page.

Protocols: Configuring FTP

For FTP you will choose a connection mode and enter timeout and password information.

Option	Description
FTP Connection Mode	**FTP transfers require two connections: a control connection (to inform the FTP server of the request for data) and a data connection (to send the data). This setting determines whether the FTP server or the Traffic Server initiates the connection.** **- PASV/PORT indicates the Traffic Server should try a PASV connection and if it fails, try a PORT connection** **- PASV ONLY mode allows the Traffic Server to initiate the connection and the FTP server to accept it. This mode is firewall friendly but some FTP servers do not support it.** **- PORT ONLY mode indicates the FTP Server will initiate the data connection and the Traffic Server will accept it.**

Cache: Configuring Cache Activation

There are three options available for configuring how cache activation is to be handled. HTTP caching (objects retrieved via HTTP) can be turned on or off, FTP caching (objects retrieved via FTP) can be turned on or off, User requests to bypass caching (server only from the Web) can be ignored or respected.

Options	Description
HTTP Caching On/Off	Tell the Traffic Server whether or not to cache objects retrieved via HTTP.
FTP Caching On/Off	Tell the Traffic Server whether or not to cache objects retrieved via FTP.
Ignore User Requests Bypass Cache	If users stipulate that their requests should not be served from the cache, ignore the stipulation.

Cache: Viewing Cache Storage

When you choose to "view" cache storage, you see a list of all cache partitions and their sizes (raw partitions may not have an associated size.

Cache: Configuring Object Freshness

You determine how fresh documents will be by setting verification on a variety of options. *Verify freshness before serving* settings tell the Traffic Server how you want to handle verification to the original server when the object has expired or if the object has no expiration date, all of the time or never. *Check objects with no expiration date* settings tell the Traffic Server how long objects without an expiration date should remain in the cache (you can set a value ranging from 15 minutes to 2 weeks).

You can also set an expiration value for FTP objects (which carry no time stamp or date information).

Cache: Configuring Variable Content Options

Some Web servers answer requests to the same URL with a variety of objects whose content can vary widely. This content may vary because of language or the different presentation styles of browsers, or it may be that the content changes at different times of the day. You can set options that prevent the caching of objects containing ? or /cgi-bin or objects that contain cookies. You can also set options for special handling of documents, specifying that if they do not match your criteria, they will not served from the cache.

Security

The Traffic Server can work inside or outside a firewall. This page allows you to configure how firewall integration will be handled and also provides restrictions to using the Traffic Manager (the graphical user interface to the Traffic Server).

Security: Controlling Access to the Traffic Manager UI

Restrict use by requiring a user ID and password.

Options	Description
Authenti-On/Off	Leave on to check admin user ID and password
Admin's ID	Specify administrator's login ID (may not include a colon).
Change Admin Password	Click this link to change the password.
Guest ID	Specify a guest login ID -- this is static for all guests.
Change Guest Password	Click this link to change the password.

Security: Configuring Firewall Integration

You configure Traffic Server integration into your firewall (inside or outside) by setting integration options. If you have no firewall or if the Traffic Server is outside of your firewall, you can leave the SOCKS flag off (and this is the default).

If your Traffic Server is inside the firewall turn the SOCKS flag on and provide the IP address of your SOCKS server, a port for the Traffic Server to connect to the SOCKS server and the related timeout information. Add a path to your SOCKS list for easy editing of IP addresses.

Routing

The routing page handles all redirection of requests.

Routing: Setting Parent Caching Options

You can point your Traffic Server at another Traffic Server to form a hierarchy of searching for requested objects. If the object is not found in the local cache, the next check is against the parent cache.

Options	Description
Parent Caching On/Off	Default is Off. If you turn parent caching on, you must provide a parent server name and port.
Name of Parent Cache	Identifies the parent cache to be used if the Traffic Server cannot find the object in its own cache. You can set multiple levels.
Parent Port	Specify a dedicated port number to be used by the Traffic Server to connect to a server containing the parent cache. Default is 8092.

Routing: Setting ICP Options

The Traffic Server supports Internet Cache Protocol (ICP). You set up a cache hierarchy and the Traffic Server will query the hierarchy for cache bits. An ICP hierarchy is not the same as HTTP parent caching. In an ICP hierarchy, ICP messages are exchanged;; in the HTTP hierarchy, HTTP requests are exchanged. In the event of a cache miss, the Traffic Server sends ICP messages asking the hierarchy members if they can serve the request. If a request is a miss for the entire ICP hierarchy, ICP returns a parent IP/proxy port to the Traffic Server which in turn forwards the HTTP request to the given IP/proxy port. The targeted parent may then resolve the request from its own cache or from the origin server.

Options	Description
ICP Enabled On/Off	Enable ICP for the Traffic Server (you must establish ICP peers).
ICP Multicast Enabled On/Off	If your Traffic Server has a multicast channel connection to its ICP peers, it can send ICP messages through multicast with this option enabled.
ICP Query Timeout	Specifies timeout (in seconds) for ICP queries.
ICP Peers	Click this link to view or modify the Traffic Server's ICP hierarchy. Hostname & Host IP: Enter host name and IP address if you know it. If you don't, leave the IP address as 0.0.0.0. Type: Enter 1 for parent cache, 2 for sibling cache, and 3 for local host (reserved for Traffic Server) Proxy Port: Traffic Server's proxy port (usually 8080) ICP Port: Enter the UDP port to be used for ICP (usually 3130)

	Multicast Member — Enter 0 if the host is not on a multicast network with the Traffic Server and 1 if it is. **Multicast IP** — Enter the multicast IP address. **Multicast TTL** — Multicast datagram - time to live: 1 - datagrams will not be forwarded beyond a single subnetwork and 2 - allows delivery of IP multicast datagrams to more than one subnet if there are more multicast routers attached to the first hop subnet.

Routing: Configuring Server Acceleration (Reverse Proxy)

You can use your Traffic Server as a server accelerator, also known as a reverse proxy. When you do this, Traffic Server receives abbreviated URLs from requesting clients, who are expecting their destination servers (for which Traffic Server is proxying) to know the full paths for their requests. The Traffic Server modifies web redirects to the servers it is accelerating, so that browsers are actually redirected back to the Traffic Server, rather than around it.

Document routing rewrite rules specify the location of content that Traffic Server is accelerating. The rules translate a URL requested by a client into one that represents the accelerated content. The step-by-step details of setting up server acceleration are included in the Solutions Workshop.

Configuring the Host Database

The host database stores the domain name server (DNS) entries of servers that the Traffic Server will contact to fulfill user requests. Configuration determines how long DNS entries will remain in the database before they are flagged as stale and refreshed. It is important to realize that these settings can have an impact on performance and/or accuracy.

You can choose to have refreshing occur only after entries become stale (before serving to users) or to continuously refresh in a background mode. You can even set negative DNS caching to remember invalid host names for the period of time you specify. You control the number of entries that can be made to this database.

Options	Description
Lookup Timeout	How long the Traffic Server will look for DNS entries.
Foreground Timeout	How long DNS entries can remain in the database before they are flagged as stale. If you set this too low it might slow response time. Too high risks accumulation of incorrect information.
Background Timeout	How long DNS entries can remain in the database before they are flagged as entries to refresh in background.

Configuring DNS

To provide DNS services, the Traffic Server uses a list of DNS servers obtained from the DNS table in your resolv.conf file. The Traffic Server always tries to connect to the first server on this list and if it is unsuccessful, it continues to through the list, entry by entry.

Configuration options allow you to specify how long the Traffic Server should wait for the DNS server to respond with an IP address. Even if the user gives up before this time is hit, the response can still be cached for subsequent use. You can also specify how many times the Traffic Server should allow a look-up before it sends back an "invalid host name" message.

Options	Description
Resolve Attempt Timeout	How long the Traffic Server should wait for the DNS server to respond with an IP address -- even if the client has been cancelled.
Number of Retries	How many times the Traffic Server should allow a look-up to fail before it abandons the look-up and sends an invalid host name message.

Logging: Configuring Event Logging

There are a number of logging configuration options which allow you to decide how much disk space will be used for your logs, how and where logs should be collated, how and when to roll logs, how to set up auto-delete to handle out-of-date logs, and what format you want you logs to be written in.

Options	Description
Log Directory	The name of the directory in which to store access logs. The name of this directory must be the same on every node in the cluster.
Log Space Limit (MB)	The maximum amount of space (in megabytes) allocated to the logging directory for log files. Be sure this space is less than the actual space available on partition containing this directory.
Log Space Headroom (MB)	Tolerance for the log space limit. If autodeletion of old logs is enabled, autodeletion is triggered when the amount of free space available in the directory is less than this headroom value.
Log Buffer Size (B)	Specifies the size in bytes of the log buffer. The log buffer improves Traffic Server performance by grouping transaction data to be written to the disk. The default value is optimized so you should not need to change this value.
Max entries Per Log Buffer.	Specifies the maximum number of entries in the log buffer. The default value is optimized so you should not need to change this value.

Logging: Configuring Log Collation

Log collation brings all logs from various nodes together when you establish a log collation server and port. If the Traffic Server cannot connect for some reason, it writes individual "orphan" log files to local disks. You will provide a name and port for this server (and the default is 8999). It's a good idea to specify a secret password in "Log Secret" to prevent any process other than the Traffic Server from writing to the log directory.

Logging: Configuring Standard Event Logs

You choose the format and names of your log files. The default for the error file is error.log, but you can rename this file. You will also choose the method for defining how the log is formatted. You have these choices:

- Squid
- Netscape Common
- Netscape Extended and Netscape Extended2

Inktomi recommends that you use one of these standard log formats for your access logs. You can enable more than one standard log file, but remember that log files consume space quickly. Samples of the various log types can be found in the Additional Materials section of this workbook.

Logging: Configuring Log Rolling and Autodeletion

Logging can take up a great deal of space so the Traffic Server comes with options to help you manage current files and historical files. You will specify guidelines for the Traffic Server to "roll" the log file, which stops it from making entries in the file and changes the file extension to .old. To keep your log directory from filling, you can also enable autodeletion, which deletes the oldest rolled log files when the space allocated to logging is nearly full. The amount of this space is determined by the headroom setting.

Options	Description
Rolling Enabled (on/off)	Enable log file rolling.
Roll Interval	How long should the Traffic Server continue entering information in log files before rolling them to .old files
Auto-Delete Rolled Files (Low Space)	Turns on autodeletion when available directory space is low.
Log Rolling Offset Hour	Time of day in the range 0-23 that specifies when rolling is to take place. If offset is 0 (midnight), and roll interval is 6 (hours), then log files will roll at midnight, 6AM, noon and 6PM.

Using Snapshots

Snapshots represent the sum of all configuration settings at a particular place in time. Options here allow you to:

- Name and Take a Snapshot
- Restore a Snapshot

Create a Snapshot before you make any changes to your configuration files. After you've created the snapshot you will notice there are new options available for you to restore or delete the snapshots you've taken.

Progress Check

By now you should be able to:

1. ***Work with the various configuration settings***
2. ***Review the changes in actual configuration files***

Unit 3 Practice Lab

Objectives for Unit 3 are to understand the various options available to you in configuring the Traffic Server and their impacts on Traffic Server performance.

1. Since you will be making changes, take a snapshot (Configure tab). Name your snapshot "post-install." Notice that after taking the snapshot, you will see new options to restore and delete your snapshot. Snapshots are captured on the system under the config/snapshots directory.

2. Select Server from Configure menu. Scroll down to the section on Auto-Configuration of browsers and create an auto-configuration file. (This file - proxy.pac - is created in the config directory.)
   (a) Check 'Hosts with non-qualified domain name' (internal requests automatically bypass the cache).
   (b) Enter the domain name (inktomi.com) to indicate internal requests -- specifying inktomi.com -- should also bypass the cache).
   (c) Uncheck 'Internal Cluster Failover' as we are not running a cluster at this time.
   (d) Leave 'Go Direct as Last Resort' checked.
   (e) Click 'Create' and notice that new options appear on this page. Choose 'View the current file' to see the auto-configuration file that was created for you.
   (f) To use this auto-config file, you will now set the "Automatic Proxy Configuration" in your browser to the auto-config port: http://{studentIP}:8083. If you aren't sure of the port number you can refer to the Tsinstall.log file. As we used defaults, this port is correct.

3. Tail your squid log once again as you visit the inktomi site (which should not be cached) and other sites like Sun, IBM or Yahoo that will be cached.

4. It is important to manage your logging. Go to the logging configuration page and scroll through the log management options. Set your log file rolling to occur every 30 minutes and make sure that auto-delete is on. This way if the space allocated to your log files is dangerously close, the oldest files will automatically be deleted to make more room.

5. Now let's take a look at the log files and a few config files to see the impact of the changes that you've made.

   (a) From the command line, move to the config directory and view the proxy.pac file. This file shows the changes from the auto-configuration. In all cases, configuration files track changes you made through Traffic Manager UI. Take a few minutes to review some of the other configuration files (particularly records.config and storage.config) in this area. Using the details provided on configuration files in this section, you can get a pretty good idea of how the configuration files work with the Traffic Manager UI. Remember that it is best to make changes to configuration files through the UI - but there are some choices that are only possible available to you through direct editing of these files.

   Whenever you make changes to configuration files, you need to activate Traffic Line to force a re-read of the files. To do this, move to the bin directory and execute Traffic Line with the –x flag: `./traffic_line -x`

   Here are a few ideas for getting started with config files. Be sure to review the information in your Traffic Server Administrator's Guide for each of these config files:

   Cache.config
   Logs.config
   Records.config
   Storage.config

   (b) Move to the logs directory and spend a few minutes looking at the logs that have been created. Check your syslog (`/var/adm/messages`) -- which you should do on a regular basis -- and the access log (which at this point is probably still squid.log unless you have already made changes).

   (c) Add another logging format through the UI and view the files.

UNIT 4: MONITORING THE TRAFFIC SERVER

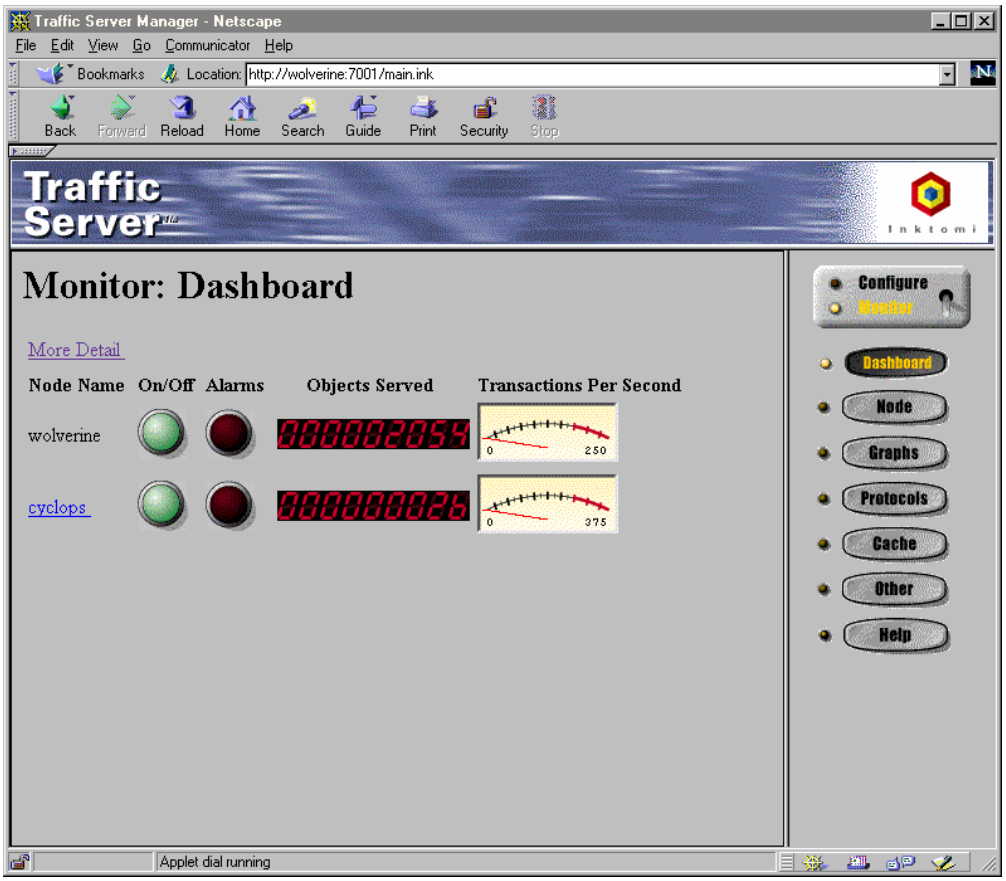
Objectives for this Unit:

- ✓ Explore Traffic Server Monitoring Features
- ✓ Learn About Various Log Formats and Log Analysis Tools
- ✓ Review the Most Important Statistics to Monitor

Monitoring includes collecting and interpreting Traffic Server performance statistics. The Traffic Server reports on performance in the form of graphs, summary statistics and raw statistics. In Monitor mode, the Traffic Server presents this information as a series of browser pages.

The Dashboard

The Dashboard page provides a view of your entire system. It displays each of your cluster nodes by name and keeps track of essential statistics for each of the nodes. With the exception o f the Dashboard and cluster information on the Node page, all statistical information pertains to a single node.



Responding to Alarms

Over time, you will develop an understanding about the many statistics for your particular Traffic Server. Should problems occur, you will be notified by alarms on the Dashboard. When you click to "resolve" the alarm, it simply means you have been notified about the warning or error condition and it does not mean you have actually taken whatever action is necessary to correct problems.

Alarms alert you of existing or potential problems for a range of conditions built into the Traffic Server. You can choose how to respond to them and can even write a script to execute certain actions for you automatically. A sample script is provided by Inktomi, called **`example_alarm_bin.sh`**, which you can modify to include how you want alarms handled. For example, you could page yourself when error conditions exist.

When an alarm is activated, you can click on it to view the page that describes the specific conditions and then take appropriate action.

Node Status

The Node status page provides performance statistics for the selected node, with details about how the overall cluster is performing. You will find a number of statistics here including document hit rates, DNS lookups and both client and server access transactions, along with cache and network information.

Graphs

There are a number of informational graphs related to Traffic Server's performance and maintenance. You choose which graphs to create when you click on the particular graph description. Graphs show performance on cache results, garbage collection, transfer rates and object sizes. You can select the format of the graph.

Protocols

When you use the Protocols page, you can obtain cluster-wide statistics for the Traffic Server's use of HTTP, NNTP, ICP and FTP protocols.

If you are using FTP in PASV (initiate connection) mode, you can compare the number of successful and failed PASV connections to number of successful and failed PORT connections to determine if time is being wasted falling back on PORT when PASV fails. Generally, PASV is preferable because it is more secure but some FTP servers do not support PASV mode.

This same page also provides statistics for NNTP for client and server performance and displays a number of operational statistics on hits, misses, refreshes and posts, as well as ICP statistics on remote requests, cache lookups (both failed and successful) and query hits.

Using the Cache Page

The Cache page statistics report cumulative and current information about connections, transactions, objects reads and writes, and document hits and misses.

The Other Page

This page reports Traffic Server functions related to logging, DNS lookups and SOCKS use. If you see more lookups on the DNS server than in the host database, you might need to increase the size of your database, or adjust database timeouts. Another place to check would be the time-out and retry settings for DNS lookups. Make these changes on the Host Database page (under Configure).

Security statistics represent cumulative activity on connections between the Traffic Server and the SOCKS server.

Logging statistics show a tally of cumulative bytes that the Traffic Server logged to local files and to remote files, as well as a total of how many log files are open at the moment of your inquiry.

Analyzing Performance Statistics

While you can monitor general performance from the Dashboard, and be notified about alarm conditions, the details live in log files. The Traffic Server uses logs to record information about itself and the state of each access or request that it processes. There are three types of logs:

- System Information Logs
- Access Request Information Logs
- Access Request Error Logs

You can use third-party log analysis tools to gather the best information from your log files. It is important to remember that Inktomi cannot provide support on these third party tools but we have included a script for the most popular of these programs (Calamaris) in your class tools directory. We have also included a program that will become part of the Traffic Server distribution with the next release of the product. MRTG allows you to see aggregate statistics in a browser window. You are welcome to take these tools with you for use at your site.

Calamaris is already distributed on the Traffic Server CD-ROM (calamaris.pl) or you can get it from http://www/detmold.netsurf.de/homepages/cord/tools/squid. Another tool, called cacheing.analysis.pl is also available on the Traffic Server CD-ROM. Others that may be of interest to you are available from NLANR. These scripts can be downloaded from http://squid.nlanr.net/Squid/Scripts. They are access-extract.pl, access-summary.pl, and squid-logs.pl.

System Information Logs

The Traffic Server sends state messages to the system log file (syslog). These messages include errors or warnings related to rolling log files, cluster communication time-outs, or Traffic Server restarts and result in alarms on the Dashboard. It provides a single repository for messages from all Traffic Server

processes. The **`syslog.conf`** configuration file (in /etc/directory) specifies where these messages are actually logged. Inktomi recommends a completely separate filesystem for logs, but this is not required. The idea is to ensure logging does not take cache space. The default location for Solaris is /var/adm/messages and the default for Digital is in /var/adm/syslog.dated/<date>/daemon.log. You will also see a continuous "heartbeat" check from the Traffic Cop process to ensure that Traffic Server is up and running -- and if it is not, to restart it using the Traffic Mom process. This check is happening every 5 minutes.

```
September 21 15:15:00 sparky traffic_cop [13318]: periodic cop
heartbeat successful
…
```

Access Request Logs

Individual access transaction requests are logged to the main logs directory. The traffic.out file is the stdout for the traffic_manager process. These access logs can be captured using several standard formats or you can create your own user-defined format. You will find the squid format in squid.log. A complete description of this format is available at http://squid.nlanr.net/Squid/FAQ/FAQ-6.html#ss6.1.

If you prefer the Netscape formats you have a choice of Netscape Common in common.log, Netscape Extended in extended.log and Netscape Extended-2 in extended2.log. The current log format (and file) is based on the choices you made when configuring your Traffic Server. To change these settings, go to the "Standard Event Log Formats" section under Traffic Manager's Configure Logging page.

There are several settings that work together to define how logging will be handled. It is important to set up your Traffic Server with each of these areas configured appropriately for your system. For instance, the log directory size limit governs how much space will be allocated to capturing logged transactions. This number is in megabytes and while the default is a low 10 megabytes, Inktomi recommends you up this limit to 100 megabytes. Another important setting has to do with log collation. When things are running smoothly, logs will be collated automatically (with this feature turned on) for all the nodes in your cluster. This can consume some cluster bandwidth and therefore impacts the performance of the cluster. If Traffic Server cannot write to the destination machine for collation, it begins creating what are called orphan log files that you will need to manually collate before you can perform any meaningful cluster-wide analysis.

In some ways, you could think of this as a sort of symphony of settings and procedures that work together to contribute to the management of your Traffic Server and your understanding of it's performance. You can monitor logs to determine how often you should roll them over to a new log file and when work should be scheduled. At the same time, you can set auto-delete features to eliminate the oldest of these files when you are in danger of running out of disk space. Again, with the symphony in mind, consider your responsibilities for analysis to determine how often you should run analysis reports prior to having logs deleted.

The log files in /inktomi/log roll to <hostname>.<from date>.<from time>-<end date>.<end time>.old at whatever roll interval you have configured, and begin this process based on your roll offset hour (another configuration setting). The defaults are to roll every six hours, beginning at midnight. Log files will also roll automatically whenever there is a server restart.

Access Transaction Log Formats
(See log.config for samples and format information)

Squid format shows time elapsed, remotehost code/status, bytes method, URL, rfc931 peerstatus or peerhost:

```
892749272 227 209.1.33.186 TCP_IMS_MISS/200 1209 GET http://www.inktomi.com/imag
es/StartNavOff.gif - DIRECT/www.inktomi.com image/gif
892749272 343 209.1.33.186 TCP_MISS/200 12038 GET http://www.inktomi.com/images/
TechTop.gif - DIRECT/www.inktomi.com image/gif
892749283 343 209.1.33.186 TCP_HIT/200 12038 GET http://www.inktomi.com/images/
TechTop.gif - NONE/- image/gif
892749292 72 209.1.33.186 TCP_CLIENT_REFRESH/304 254 GET http://www.inktomi.com/
images/AddrBlock.gif - DIRECT/www.inktomi.com image/gif
traffic_cop heartbeat
892749300 3 127.0.0.1 TCP_MISS/200 1829 GET http://127.0.0.1:8083/synthetic.txt
- DIRECT/127.0.0.1 text/plain
```

Netscape Common logfile format shows remotehost, rfc931, authuser ,[date], "method URL," status, and bytes. See (http://www.w3.org/Daemon/User/Config/Logging.html#common-logfile-format) for more information on all of Netscape's log formats.

```
netscape-common
209.1.33.186 - - [16/Apr/1998:11:18:57 +0700] "GET http://www.aol.com/gr/New2.gi
f 1.0" 200 104
209.1.33.186 - - [16/Apr/1998:11:18:58 +0700] "GET http://ads.web.aol.com:80/con
tent/3819833189/29532/SHOP4.GIF 1.0" 200 1995
209.1.33.186 - - [16/Apr/1998:11:15:35 +0700] "GET http://www.inktomi.com/images
/OnSuppNavOff.gif 1.0" 304 0
traffic_cop heartbeat
127.0.0.1 - - [16/Apr/1998:11:15:44 +0700] "GET http://127.0.0.1:8083/synthetic.
txt 1.0" 200 1620
```

Netscape Extended format:

```
netscape-ext
209.1.33.186 - - [16/Apr/1998:11:30:18 +0700] "GET http://www.digital.com/ 1.0"
200 13719 200 13719 0 0 269 173 364 122 1
```

Netscape-ext-2 format:

```
netscape-ext-2
209.1.33.186 - - [16/Apr/1998:11:35:05 +0700] "GET http://www.inktomi.com/images
/TechNavOff.gif 1.0" 304 0 304 0 0 0 402 254 497 72 0 DIRECT FIN FIN UP-TO-DATE
209.1.33.186 - - [16/Apr/1998:11:35:05 +0700] "GET http://www.inktomi.com/ 1.0"
304 0 304 0 0 0 354 254 449 72 0 DIRECT FIN FIN UP-TO-DATE
```

Error Logs

The Traffic Server will retry 6 times on a server giving connect errors before sending a respons error 502 "Connection Refused." Other response errors you may see are Bad Header (500), Unknown host (500), etc. The Traffic Server Administration Guide has an entire appendix on the various Traffic Server and HTTP error messages that will be captured.

```
Traffic Server 1.1.0 rc17 [SSL] (build 0404) [08/Apr/1998:12:36:23 +0700] cyclops:
CONNECT: could not connect to 207.200.77.4 2 for 'http://search.netscape.com/search-
bin?NS-search-page=result' (marked address invalid)
[08/Apr/1998:12:38:23 +0700] cyclops: CONNECT: could not connect to 207.200.77.4
2 for 'http://search.netscape.com/search-bin?NS-search-page=result'
(still retaining address)
```

Ad Hoc Queries of Error Logs

You can get basic statistics using grep on your log files. Here are some sample commands:

All activity from a client workstation

```
grep 206.205.249.56 squid.log*
```

Status 500, 502 or 504 errors to a client

```
grep 206.205.249.56 error.log* | grep RESOLV
```

All pages loaded from an origin server

```
grep www.inktomi.com squid.log* | grep DIRECT
grep www.inktomi.com squid.log* | grep DIRECT | wc -l (just count pages
```

Watch current activity on Traffic Server

```
curlog=`ls squid.log* | grep -v "\.old"`; export curlog
tail -f $curlog
```

Or to put a similar command in a shell (csh)

```
setenv curlog `ls squid.log* | grep -v "\.old"`
tail -f $curlog
```

Using Calamaris to Report on Traffic Server

The next few pages show the kind of information you can retrieve using Calamaris. There is a script in the class tools directory that generates this particular report.

```
# Request peak per Protocol
     sec  peak begins at     min   peak begins at      hour   peak begins at
--- ---- ----------------- ----- ----------------- ------- -----------------
UDP    0                       0                         0
TCP   28 16.Apr 98 13:41:10    48 16.Apr 98 13:40:40     598 16.Apr 98 13:17:58
--- ---- ----------------- ----- ----------------- ------- -----------------
ALL   28 16.Apr 98 13:41:10    48 16.Apr 98 13:40:40     598 16.Apr 98 13:17:58
# TCP-Request State                 Request     %    kByte      %   sec  KB/sec
---------------------------------- ------- ------ -------- ------ ---- -------
HIT                                     43   5.72       82   5.33 0.17   10.78
 TCP_HIT                                42   5.59       79   5.15 0.09   20.38
 TCP_REFRESH_HIT                         1   0.13        2   0.18 3.73    0.76
MISS                                   530  70.48     1372  88.92 0.39    6.59
 TCP_MISS                              521  69.28     1362  88.31 0.38    6.80
 TCP_CLIENT_REFRESH                      6   0.80        6   0.44 1.07    1.06
 TCP_IMS_MISS                            2   0.27        0   0.02 0.08    2.08
 TCP_REFRESH_MISS                        1   0.13        2   0.14 1.35    1.59
ERROR                                  179  23.80       88   5.76 0.00  565.00
 ERR_INVALID_REQ                       177  23.54       88   5.76 0.00 3828.34
 ERR_CLIENT_ABORT                        2   0.27        0   0.00 0.06    0.00
---------------------------------- ------- ------ -------- ------ ---- -------
Sum                                    752            1543        0.28    7.14

# external Fetches
Type Neighbor                      Request     %    kByte      %   sec  KB/sec
---------------------------------- ------- ------ -------- ------ ---- -------
DIRECT                                 530 100.00     1372 100.00 0.39    6.59
---------------------------------- ------- ------ -------- ------ ---- -------
Sum                                    530            1372        0.39    6.59

# Request-Peak by 2ndlevel-domain       Request     %    kByte      %   Hit-%
---------------------------------------- ------- ------ -------- ------ ------
127.0.0.*                                    448  59.57      801  51.94   0.00
                                             177  23.54       88   5.76   0.00
*.abcnews.com                                 38   5.05       70   4.55  52.63
*.schwab.com                                  31   4.12       58   3.82  45.16
*.alteon.com                                  29   3.86      362  23.51   0.00
*.news.com                                    14   1.86       76   4.93  64.29
*.inktomi.com                                  6   0.80       33   2.15   0.00
*.bigcharts.com                                2   0.27        3   0.21   0.00
206.146.143.*                                  2   0.27        6   0.43   0.00
---------------------------------------- ------- ------ -------- ------ ------
Sum                                          752 100.00     1543 100.00   5.72

# Request-Peak by toplevel-domain       Request     %    kByte      %   Hit-%
---------------------------------------- ------- ------ -------- ------ ------
unresolved                                   450  59.84      808  52.37   0.00
                                             177  23.54       88   5.76   0.00
*.com                                        125  16.62      646  41.88  34.40
---------------------------------------- ------- ------ -------- ------ ------
Sum                                          752 100.00     1543 100.00   5.72

# requested content-type                Request     %    kByte      %   Hit-%
---------------------------------------- ------- ------ -------- ------ ------
text/plain                                   450  78.53      803  55.23   0.00
image/gif                                     91  15.88      370  25.46  43.96
text/html                                     21   3.66      159  10.97   9.52
image/jpeg                                     8   1.40      118   8.15  12.50
```

```
none/-                                    2   0.35        1   0.08   0.00
application/octet-stream                  1   0.17        1   0.11   0.00
----------------------------------------- ------- ------ -------- ------ ------
Sum                                     573 100.00     1455 100.00   7.50


# TCP-Requests                          Request     %    kByte      %   Hit-%
----------------------------------------- ------- ------ -------- ------ ------
localhost                                   448  59.57      801  51.94   0.00
femmes.inktomi.com                           49   6.52      472  30.59  18.37
nova                                         80  10.64      181  11.79  42.50
belly.inktomi.com                            44   5.85       22   1.44   0.00
stones.inktomi.com                           22   2.93       10   0.71   0.00
buffett.inktomi.com                          15   1.99        7   0.49   0.00
parker.inktomi.com                           15   1.99        7   0.48   0.00
filter.inktomi.com                           14   1.86        7   0.46   0.00
dao.inktomi.com                              14   1.86        6   0.44   0.00
tressa.inktomi.com                           12   1.60        6   0.41   0.00
amy.inktomi.com                              10   1.33        4   0.32   0.00
...
vega                                          2   0.27        0   0.06   0.00
dpierce.inktomi.com                           1   0.13        0   0.04   0.00
cranberry.inktomi.com                         1   0.13        0   0.03   0.00
----------------------------------------- ------- ------ -------- ------ ------
Sum                                         752 100.00     1543 100.00   5.72
```

Using a cacheing.analysis Report

This report sorts by URL and examines operation sequences. Be careful about keeping these potentially large log files around, especially since this script can be slow and can exhaust memory.
Strip synthetic.txt references using grep -v. This is just a small sample of the actual file.

cache.analysis.pl squid.log >save.analysis

```
http://www.inktomi.com/images/ProdPartNavOff.gif :
  892749160 10:52 TCP_IMS_MISS/200 ( 0.1s transtime) (Possibly first ever request.)
  892749292 10:54 TCP_CLIENT_REFRESH/3 ( 0.1s transtime) ( 2.2m since last)
  892750321 11:12 TCP_HIT/200 ( 0.0s transtime) (17.1m since last)
  892752475 11:47 TCP_HIT/200 ( 0.0s transtime) (35.9m since last)
http://www.cnet.com/Images/newbump.gif :
  892752272 11:44 TCP_MISS/200 ( 0.2s transtime) (Possibly first ever request.)
  892752295 11:44 TCP_CLIENT_REFRESH/3 ( 0.1s transtime) (23.0s since last)
http://www.cnet.com/Ads/Media/Images/micron.9804_portyellow.portal.gif :
  892752273 11:44 TCP_MISS/200 ( 0.7s transtime) (Possibly first ever request.)
  892752296 11:44 TCP_CLIENT_REFRESH/3 ( 0.5s transtime) (23.0s since last)
```

Progress Check

By now you should be familiar with:

1. ***Monitoring features and options***
2. ***Log file features and options***
3. ***Log analysis tools you can use to monitor overall performance of your Traffic Server***

Unit 4 Practice Lab

Objectives for Unit 4 are to understand the various options available to you in monitoring the Traffic Server, especially in terms of logging and the log analysis tools you can use to ensure a well-performing application.

1. Review each of the pages and options for monitoring the Traffic Server from the Traffic Manager.

2. Use the calamaris_report script (in your class tools directory) to analyze your server's performance. There is a copy of a squid log in this directory. It has a fair amount of information in it to give you a good look at the value of the calamaris report.

   ```
   calamaris.pl squid.log >calamaris.report
   ```

3. View the report, then run it once more against the actual Traffic Server log in your 2.1/logs directory.

4. Try the different settings for logging formats and review the log files created. When finished, reset to the Squid format.

5. Practice making changes and seeing the results. In some cases, do the wrong thing on purpose, like change your logging directory to a non-existent one – (but be sure to correct errors right after you cause them so you don't get yourself into too much trouble. Modify values in log sizes to force an error that causes logging to stop.

   Make a few other changes like turning your server on and off and checking contents of the log files or configuration files as appropriate.

UNIT 5: MAINTENANCE, PERFORMANCE AND TROUBLESHOOTING

Objectives for this Unit:

- ✓ Review Warning and Error Messages
- ✓ Get Tips from Inktomi Pros on Maintaining Your Traffic Server

Maintaining Your Traffic Server

It is important to realize that your general network configuration outside the Traffic Server is every bit as important as the configuration inside the Traffic Server. Maintaining the Traffic Server is primarily a matter of monitoring performance, tuning configuration settings, responding quickly to warnings and errors and scaling your application should the need arise. You may need to recover log files in the event the log collation server fails, and occasionally may need to respond to users about error messages directed to their browsers. For this reason, this section covers the various error and warning messages that you will see in alarms or log files, and where possible, ideas for correcting situations that may arise.

Monitoring the Dashboard is an ongoing process. Any serious problems will turn on the "Alarm" light. Detailed messages will also be seen in system log files. You should observe alarm message and dismiss them, realizing that the purpose of the alarm is to notify you of a condition requiring your attention and that dismissing the alarms does not necessarily coincide with actually correcting the error condition. If you have trouble figuring out what the error is, contact Technical Support. Keep in mind that there is an online support center with many technical notes for your use. Inktomi's Solutions List can be found at http://www.support1.inktomi.com.

Network Issues

To check for routing or network configuration problems, you can use ping, traceroute, or other network diagnostic tools and get some help from your network administrator. Here are some errors you might see, and what they mean.

Message	Description
Proxy cannot resolve host names	Missing or incorrect nameserver entries in /etc/resolv.conf. Nameserver is not operating correctly. Edit /etc/resolv.conf to check and run nslookup to verify name service operation.
Clients cannot connect to proxy	Routing or network configuration problems exist between clients and proxy
Proxy cannot connect to servers	Routing or network configuration problems exist between lients and proxy
Slow service or connect times	Routing or network configuration problems.

One good way to do a quick proxy test is to simply telnet to the proxy port. Here is a basic proxy test

that you may find helpful:

```
storm:~ [67] % telnet cyclops 8180
Trying 209.1.32.98...
Connected to cyclops.inktomi.com.
Escape character is '^]'.
GET www.inktomi.com HTTP/1.0
HTTP/1.0 200 OK
Server: Microsoft-IIS/3.0
Date: Mon, 15 Jun 1998 17:53:02 GMT
Content-Type: text/html
Accept-Ranges: bytes
Last-Modified: Fri, 12 Jun 1998 17:18:04 GMT
Content-Length: 7445
Via: 1.0 cyclops (Traffic-Server/1.1.2 [O])
<html>
<head>
</body>
</html>
Connection closed by foreign host.
```

Another tool that may be helpful is the batch WWW document reader called "wget" which is configured in ~/.wgetrc. You can get one URL or get several from a text file list of URLs. This can be used to accomplish a cache prefetch.

```
Using the wget batch WWW document reader (configure with ~/.wgetrc):
ftp_proxy = cyclops:8090   # hostname:port for FTP proxy
http_proxy = cyclops:8090  # hostname:port for HTTP proxy
use_proxy = on             # use named proxies; set “off” to go direct to server
recursive = off            # page loading recursion on/off
reclevel = 2               # depth of recursion
numtries = 1               # number of retries if HTTP GET fails
nowrite = off              # off/on for do/don’t write received documents to disk
robots = off               # use robots files on web servers
server_response = on       # show server response headers
```

Alarm Messages

Message	Description
Access logging suspended - configured space allocation exhausted.	Space allocated to access log files is full. You must either enlarge space or delete some log files. To prevent this, roll log files more frequently or enable the autodelete feature.
[Rollback:Rollback] Config file is read-only:<filename>	Go to the /config directory and check the file’s permissions - change as necessary.

[Traffic Manager] Mgmt <==>Proxy conn. closed	**This is an informational message, letting you know the traffic_server process was down. You would see this whenever a restart is issued.**
Traffic Server failed to parse line <line number> of the logging config file <filename>	**Check your custom log configuration file. There are probably syntax errors.**

Monitoring Traffic Server Logs

The Traffic Server maintains three different logging systems. You will find your ***system logs***, which contain system errors, warnings, and information messages logged by the syslog facility in the appropriate directory, by platform. On Solaris: this will be `/var/adm/messages` and on Digital UNIX, `/var/adm/syslog.dated<date>/daemon.log`. ***Access Logs*** will be found in the Traffic Server's logging directory. These are the individual transactions as they are processed and are the files you need to be sure to roll regularly. Be sure to consider using auto-delete features to avoid filling up space. This feature eliminates the oldest files first, whenever your log files are getting dangerously close to the limits you have set for log files. You will want to be sure to analyze your log files before an auto-delete, to make sure you have the information you need to manage your server. Tools like Calamaris or MRTG are great for getting at aggregate information. ***Access Error Logs*** identify problems that arise when processing transactions and they use a standard error log format.

Error Messages: Server Status

Message	**Description**
Note: machine down <IP address>	**The machine with the given IP address is down.**
Note: machine up, <IP address>, protocol version=<X.Y>	**The machine with the given IP address and protocol version is up.**
Warning: Unable to accept cluster connections on port: <cluster port number>	**Call Technical Support.**
Warning: connect by disallowed client <IP IPaddress>, closing	**The specified client is not allowed to connect to the Client allowed to connect to Traffic Server proxy. The client IP address is not listed in the ip_allow.config file.**
ProcessFatal: accept port is not between 1 and 65535. Please check configuration.	**The port specified in records.config for accepting incoming HTTP requests is not valid.**

Error Messages: Congestion

Message	Description
Note: Network congestion to <IP address> encountered, reverting to proxy only mode	Traffic Server is too congested to cache. This is "graceful degradation."
Note: Network congestion to <IP address> cleared, reverting to cache mode	Congestion is cleared and cache capability has returned.
Warning: Did <this amount> of backup. Still to do <remaining amount>.	Means that congestion is approaching.

Error Messages: Caching

Message	Description
Warning: No storage available. Cache disabled.	There is no cache storage available. There is a configuration or hardware problem. Check for other syslog messages that give more specific information.
Warning: cache read error	By itself, call Tech Support. With other warnings about cache segments, there is a a disk problem. Replace disk as appropriate or troubleshoot with Tech Support staff.
Note:\Vary: <header field> -- object not served from cache	Document content varies on header fields so cached copy is not being served to client.
Warning: Unable to read cache segment	Comes from garbage collector Segment may be corrupt when it cannot read a cache or there may be a disk error. Generally marks the segment as corrupt.

Error Messages: Logging

Message	Description
Warning: missing field for field marker	Error in reading a log buffer.
Warning: can’t open config file <filename> for reading custom formats	Custom logging was enabled, but can't find logs.config file.

Message	Description

Note: rolled file <filename> already exists, attempting version <version>	**Attempting to roll over an existing file, so roll is being changed.**
Warning: could not rename <log filename> to <rolled log filename>	**System error in renaming log file during roll.**
Warning: <log file> error: <error number>	**Generic logging error.**
Warning: unable to open logfile <filename>, errno=<error number>	**Can't open the log file.**
Warning: log format symbol <symbol name> not found	**Custom log format references a field that doesn't exist.**

Error Messages: Server Versions

Message	Description
Warning: Different clustering minor versions <version 1, version 2> for node <IP address> continuing	**Incompatible software versions are causing a problem.**
Warning: Bad cluster major version range <version 1 - version 2> for node <IP address>	**Incompatible software versions are causing a problem.**

Recovery

The Traffic Server recovers from all types of failure automatically, except failures that write to your log collation server. If your system crashes from hardware or software failures, the Traffic Server will automatically recovers its caches and its host database which means that all document cache contents are automatically recovered and that all DNS cache contents are automatically recovered.

When the Traffic Server cannot write to the log collation server, it creates orphan log files on the local disks, which you will need to collate manually. Step-by-step procedures are included in your Administrator's Guide.

Modifying the Cache

Many maintenance commands are executed from the command line, as the administrator. To modify the cache -- adding or removing capacity, you must edit the storage.config file. Do this very carefully as mistakes can render the cache unusable and cause the Traffic Server to run in "proxy only" mode. After you have reconfigured the cache, use the clear command to remove all data in the Traffic Server's cache. (The Traffic Server must be taken down before the clear is issued.). Plan ahead to minimize the loss of existing cached documents. Be sure to take a snapshot first. Step-by-step procedures are included in the Administrator's Guide.

For detailed performance tuning information, consult your system vendor's documentation or system-specific reference works such as "Sun Performance and Tuning" by Adrian Cockroft and Richard Pettit