Using the Transaction Logs

This chapter explains how to use the transaction logs and contains the following sections:

Transaction logs allow administrators to view the traffic that has passed through the Content Engine. Typical fields in the transaction log are the date and time when a request was made, the URL that was requested, whether it was a cache hit or a cache miss, the type of request, the number of bytes transferred, and the source IP address.

Understanding Transaction Log Formats

In ACNS 5.x software, the user can choose among Squid, Extended Squid, Apache, or customized log formats for the transaction log.

Squid-Style Transaction Logging

The Squid-style log format is the default format for transaction logging in the Content Engine. The Squid log file format used is the native log file format associated with the Squid-1.1 *access.log* file format. For details on the Squid-1.1 native log file format, refer to the Squid documentation “Frequently Asked Questions,” section 6.6, access.log heading at the following URL:

http://wiki.squid-cache.org/SquidFaq/SquidLogs

The Squid log file format is as follows:

time elapsed remotehost code/status bytes method URL rfc931 peerstatus/peerhost type

A Squid log format example looks like this:

1012429341.115 100 172.16.100.152 TCP_REFRESH_MISS/304 1100 GET http://www.cisco.com/images/homepage/news.gif - DIRECT/www.cisco.com -

Extended Squid Log Format

The Extended Squid format logs the associated username for each record in the log file, in addition to the fields logged by the Squid-style format, and is used for billing purposes. In this format the Rfc931 field associated with the Squid format is used to log the authorized user. This field always contains a "-" (dash) if no user information is available.

An Extended Squid-style log format example looks like this:

1012429341.115 100 172.16.100.152 TCP_MISS/302 184 GET http://www.cisco.com/cgi-bin/login myloginname DIRECT/www.cisco.com

Apache-Style Transaction Logging

The Apache format is the Common Log File (CLF) format defined by the World Wide Web Consortium (W3C) working group. This format is compatible with many industry-standard log tools. For more information, see the W3C Common Log Format website at the following URL:

http://www.w3.org/Daemon/User/Config/Logging.html.

The Apache-style log file format is as follows:

remotehost rfc931 authuser date request status bytes

An Apache-style log file format example looks like this:

172.16.100.152 - - [Wed Jan 30 15:26:26 2002]
"GET/http://www.cisco.com/images/homepage/support.gif HTTP/1.0" 200 632

Custom Format Transaction Logging

The **transaction-logs format custom** command allows you to use a log format string to log additional fields that are not included in the predefined native Squid format, the Extended Squid format, or the Apache CLF format. The log format string is a string that can contain the tokens listed in Table 19-1 and that mimics the Apache log format string. The log format string can contain literal characters that are copied into the log file. Double backslashes (\\) can be used to represent a literal backslash, and a backslash followed by a single quote (\') can be used to represent a literal single quote. A literal double quote cannot be represented as part of the log format string. The control characters \t and \n can be used to represent a tab and a new line character, respectively.

Table 19-1 lists the acceptable format tokens for the log format string. The "..." portion of the format tokens shown in this table represents an optional condition. This portion of the format token can be left blank, as in %a. If an optional condition is included in the format token and the condition is met, then what is shown in the Value column of Table 19-1 is included in the transaction log output. If an optional condition is included in the format token but the condition is not met, the resulting transaction log output is replaced with a hyphen (-). The form of the condition is a list of HTTP status codes, which may or may not be preceded by an exclamation point (!). The exclamation point is used to negate all of the status codes that follow it, meaning that the value associated with the format token is logged if none of the status codes listed after the ! match the HTTP status code of the request. If any of the status codes listed after the ! match the HTTP status code of the request, then a hyphen (-) is logged.

For example, "%400,501{User-Agent}i" logs the User-Agent header value on 400 errors and 501 errors (Bad Request, Not Implemented) only, whereas "%!200,304,302{Referer}i" logs the Referer header value on all requests that did not return a normal status.

The custom format currently supports the following request headers:

- User-Agent
- Referer
- Host
- Cookie

The output of each of the following Request, Referer, and User-Agent format tokens specified in the custom log format string is always enclosed in double quotation marks in the transaction log entry:

%r

%{Referer}i

%{User-Agent}i

The %{Cookie}i format token is generated without the surrounding double quotation marks because the Cookie value itself can contain double quotes. The Cookie value can contain multiple attribute-value pairs that are separated by spaces. When you use the Cookie format token in a custom format string, we recommend that you position it in the last field of the format string so that the Cookie format token can be more easily parsed by transaction log reporting tools. Alternatively, if you use the format token string "\'%{Cookie}i\'", the Cookie header can be surrounded by single quotes.

The following command can be entered to generate the well-known Apache Combined Log Format:

transaction-logs format custom "[%{%d}t/%{%b}t/%{%Y}t:%{%H}t:%{%M}t:%{%S}t %{%z}t] %r %s %b %{Referer}i %{User-Agent}i"

The following transaction log entry example in the Apache Combined Format is configured by using the preceding custom format string:

[11/Jan/2003:02:12:44 -0800] "GET http://www.cisco.com/swa/i/site_tour_link.gif HTTP/1.1" 200 3436 "http://www.cisco.com/" "Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)"

Table 19-1 Custom Format Log Format String Values

Format Token	Value
%...a	IP address of the requesting client.
%...A	IP address of the Content Engine.
%...B %...b	Bytes sent, excluding HTTP headers.
%...c	Connection status when response is completed, where: X = Connection was aborted before the response was completed. + = Connection can be kept alive after the response is sent. – = Connection is closed after the response is sent.
%...f	Filename.
%...h	Remote host (IP address of the requesting client is logged).
%...H	Request protocol.
%...{Foobar}i	Contents of Foobar: header lines in the request that is sent to the server. The value of Foobar can be one of the following headers: User-Agent, Referer, Host, or Cookie.
%...l	Remote log name. Not implemented on the Content Engine, so a hyphen (-) is logged.
%...m	Request method.

Table 19-1 Custom Format Log Format String Values (continued)

Format Token	Value
%...p	Canonical port of the server servicing the request. Not applicable on the Content Engine, so a hyphen (-) is logged.
%...P	Process ID of the child that serviced the request.
%...q	Query string (which is preceded by a ? if a query string exists; otherwise, it is an empty string).
%...r	First line of the request.
%...s	Status. The translog code always returns the HTTP response code for the request.
%...t	Time in common log time format (or standard English format).
%...{format}t	Time in the form given by the format token specified in Table 19-2.
%...T	Time consumed to serve the request in seconds (a floating point number with 3 decimal places).
%...u	Remote user.
%...U	URL path requested, not including query strings.
%...v %...V	Value of the host request header field reported if the host appeared in the request. If the host did not appear in the host request header, the IP address of the server specified in the URL is reported.

Table 19-2 specifies the format token for the date and time of the format token %...{format}t listed in Table 19-1.

Table 19-2 Format Token for Date and Time

Format Token	Value
%a	Abbreviated weekday name.
%A	Full weekday name.
%b	Abbreviated month name.
%B	Full month name.
%c	Date and time representation.
%C	Century number (year/100) as a 2-digit integer.
%d	Day of the month as a decimal number (01–31).
%D	Equivalent to %m/%d/%y. (Note that in countries other than the USA, the form %d/%m/%y is commonly used. This means that in international context this format is ambiguous and should not be used.)
%e	Like %d, the day of the month as a decimal number, but a leading zero is replaced by a space.
%G	ISO 8601 year with the century as a decimal number. The 4-digit year corresponding to the ISO week number. (See %V.) This has the same format and value as %y, except that if the ISO week number belongs to the previous or next year, that year is used instead.
%g	Like %G, but without century; that is, with a 2-digit year (00–99).
%h	Equivalent to %b.

Table 19-2 Format Token for Date and Time (continued)

Format Token	Value
%H	Hour as a decimal number using a 24-hour clock (00–23).
%I	Hour as a decimal number using a 12-hour clock (01–12).
%j	Day of the year as a decimal number (001–366).
%k	Hour (24-hour clock) as a decimal number (0–23); single digits are preceded by a blank. (See also %H.)
%l	Hour (12-hour clock) as a decimal number (1–12); single digits are preceded by a blank. (See also %I.)
%m	Month as a decimal number (01–12).
%M	Minute as a decimal number (00–59).
%n	New line character.
%p	Either AM or PM according to the given time value, or the corresponding strings for the current locale. Noon is treated as pm and midnight as am.
%P	Like %p but in lowercase: am or pm or a corresponding string for the current locale.
%r	Time in a.m. or p.m. notation. This is equivalent to '%I:%M:%S %p.'
%R	Time in 24-hour notation (%H:%M). For a version including the seconds, see %T below.
%s	Number of seconds since the epoch; that is, since 1970-01-01 00:00:00 UTC.
%S	Second as a decimal number (00–61).
%t	Tab character.
%T	Time in 24-hour notation (%H:%M:%S).
%u	Day of the week as a decimal (1–7), Monday being 1. (See also %w.)
%U	Week number of the current year as a decimal number (00–53), starting with the first Sunday as the first day of week 01. (See also %V and %W.)
%V	ISO 8601:1988 week number of the current year as a decimal number (01–53), where week 1 is the first week that has at least 4 days in the current year, and with Monday as the first day of the week. (See also %U and %W.)
%w	Day of the week as a decimal (0–6), Sunday being 0. See also %u.
%W	Week number of the current year as a decimal number (00–53), starting with the first Monday as the first day of week 01.
%x	Date representation without the time.
%X	Time representation without the date.
%y	Year as a decimal number without a century (00–99).
%Y	Year as a decimal number, including the century.
%z	Time zone as hour offset from GMT. Required to emit RFC 822-conformant dates (using "%a, %d %b %Y %H:%M:%S %z").
%Z	Time zone or name or abbreviation.
%%	Literal % character.

W3C Customizable Logging Format

To provide more flexibility in logging, ACNS software supports W3C Customizable Logging Format apart from the fixed formats like Apache Common Log Format (CLF), Squid, and Extended Squid formats.

W3C customizable Logging Format supports a set of format tokens that exposes the underlying Translog (Transaction Logs) tokens. The W3C Customizable Logging Format is limited in that it was defined from the HTTP web server perspective and does not offer certain web cache-specific custom options such as those supplied by the fixed Squid format. Consequently, additional format tokens that are extensions to the W3C Customized Logging Format were added (in the ACNS 5.3 software release) to support additional Cisco and Squid-like customized logging fields. These format tokens provide support for Squid-like logging format from within the W3C customizable token set.

ACNS 5.5 software supports the following transaction logging formats:

- Support for the Extended Squid-equivalent internal tokens that were not supported by the W3C format
- Support for an additional hierarchy token that treats a configured HTTP outgoing proxy ("http outgoing-proxy') as a Squid-style "DEFAULT_PARENT" hierarchy event

ACNS 5.5 software includes the following special token sequence for the W3C Customizable Logging Format:

%...{}C

The "..." is optional. If specified, it can be a sequence of conditional HTTP response codes separated by commas. The "C" is an uppercase "C" and defines the extended customizable behavior token set, for which tokens are defined by the directive, which is a two-character token directive.

See Table 19-3 for a list of existing and new directives from the Extended Squid format, which are not currently supported by the W3C definitions, but are supported in ACNS 5.5 software.

Table 19-3 Translog Token Directives

Format Token	Value
%...{es}C	Current time presented as the number of seconds that have elapsed since the Epox (Jan. 1st. 1970).
%...{em}C	Current number of milliseconds that have elapsed since the Epox (Jan. 1st. 1970).
%...{te}C	Number of milliseconds that have elapsed until the request was completed.
%...{rd}C	Squid-like cache-status code string (for example, TCP_HIT and TCP_CLIENT_REFRESH_MISS).
%..{cs}C	Number of bytes sent to the client (including the protocol headers).
%...{rh}C	Strict Squid-style hierarchy as it applies to the Content Engine.
%...{rH}CE	Extended Squid-style hierarchy. Same as "%...{rh}C" except when an outgoing-proxy is explicitly defined and is used to satisfy a request, and then the "DEFAULT_PARENT/proxy_ip_addess" is logged instead of the "DIRECT/origin_server_ip_address."
%...{rt}C	Mime-Type of the object in the response, as specified by any protocol headers which define such.

Table 19-3 Translog Token Directives (continued)

Format Token	Value
%...{ru}C	URL being requested, including any additional query strings.
%...{as}C	Application specific information. Certain request handling applications might attempt to log a certain string here, which is supported as part of the Squid format specification. For example, SmartFilter URL filtering will log information where this token sequence is used.

In addition to the tokens listed in Table 19-3, you can condense multiple "%...{xx}C" style tokens into a single embedded token sequence within the %...{xx}C style. To condense multiple style tokens into a single embedded token sequence, you must specify multiple tokens within the {} braces and prefixing each token with the `%' symbol. For example,

%{rh}C %{rt}C %{as}C

can be re-expressed in a condensed embedded token format as the following:

%{%rh %rt %as}C.

The command line syntax will accept single tokens represented as

%{%rh}C

and

%{rh}C

as equivalents.

Any character that is not part of an embedded token sequence (for example, the space character) is repeated verbatim in the output file.

The above set of tokens allow you to configure an extended Squid-like format line within the W3C Customizable Logging format specification. For example:

"%{es}C.%{em}C %{te}C %a %{rd}C/%s %{cs}C %m %{ru}C %u %{rh}C %{rt}C %{as}C"

The following is an example of a Extended Squid-like format that specifies that user-readable time-stamps are used instead of the Squid "seconds-since-epoch" time-stamp format, and that a configured out-going proxy (as specified by "%...{rH}C") is logged:

"[%{%d/%b/%Y:%H:%M:%S %z}t] %{te}C %a %{rd}C/%s %{cs}C %m %{ru}C %u %{rH}C %{rt}C %{as}C"

Unknown or unsupported translog tokens are logged within the log file as the characters that made up the token. For example, "%{xy}C" is logged into the log file as "xy." All characters outside of a token specification sequence are repeated verbatim within the log file.

Transaction Logging and NTLM Authentication

If your device is configured for NT LAN Manager (NTLM) authentication and uses Apache-style or Extended Squid-style format, you can record the Windows domain name and username in the "authenticated username" field of the transaction log. If the domain name is available, both the domain name and the username are recorded in the "authenticated username" field, in the form domain\username. If only the username is available, only the username is recorded in the "authenticated username" field. If neither a domain name nor a username is available, a "-" (hyphen) is recorded in the field.

Usage Guidelines for Log Files

This section provides some guidelines for working with log files.

Understanding Working Logs

Depending upon where the sysfs is mounted, transactions are logged to a working log on the local disk in one of these files:

- /local1/logs/working.log
- /local2/logs/working.log

Depending upon where the sysfs is mounted, the following log files are logged to a working log on the local disk as follows:

- WMT logs are logged to a working log on the local disk in one of these files:
  - /local1/logs/export/working.log
  - /local2/logs/export/working.log
- RealSubscriber logs are logged to a working log on the local disk in one of these files:
  - /local1/logs/real-subscriber-logs/working.log
  - /local2/logs/real-subscriber-logs/working.log
- Cisco Streaming Engine logs are logged to a working log on the local disk in one of these files:
  - /local1/logs/cisco-streaming-engine/working.log
  - /local2/logs/cisco-streaming-engine/working.log
- RealProxy logs are logged to a working log on the local disk in one of these files:
  - /local1/logs/real-proxy/working.log
  - /local2/logs/real-proxy/working.log

Archive Working Log

You can specify the interval at which the working log should be cleared by moving the data to an archive log. The archive log files are located on the local disk in the directory /local1/logs/ or /local2/logs/, depending upon where the sysfs is mounted.

Because multiple archive files are saved, the filename includes the time stamp when the file was archived. Because the files can be exported to an FTP/SFTP server, the filename also contains the IP address of this Content Engine.

The archive file name use this format:

celog_IPADDRESS_YYYYMMDD_HHMMSS.txt.

Sanitizing Transaction Logs

You can disguise the IP address and usernames of clients in the transaction log file. The default is that transaction logs are not sanitized. A sanitized transaction log disguises the network identity of a client by changing the IP address in the transaction logs to 0.0.0.0.

Exporting Log Files

To facilitate the postprocessing of cache log files, you can export transaction logs to an external host. This feature allows log files to be automatically exported by FTP to an external host at configurable intervals. The username and password used for FTP are configurable, as is the directory to which the log files are uploaded.

The log files automatically have a filename that uses this format:

<type>_<ipaddr>_yyyymmdd_hhmmss.txt

where

- *<type>* represents the type of log file with *celog* for cache logs such as HTTP, HTTPS, and FTP, and *mms_export* for Windows Media Technologies (WMT) logs.
- *<ipaddr>* represents the Content Engine IP address.
- *yyyymmdd_hhmmss* represents the date and time when the log was archived for export.

Note For WMT logs, there is no .txt extension in the filename.

Exporting Transaction Logs to External FTP Servers

To export transaction logs to an FTP server, you must first enable exporting of transaction logs and then configure the FTP or secure FTP (SFTP) server parameters. This feature can support up to four FTP servers. The following information is required for each target FTP server:

- Server IP address or the host name

  The Content Engine translates the host name with a DNS lookup and then stores the IP address in the configuration.

- FTP user login and user password
- Path of the directory where transferred files are written

  Use a fully qualified path or a relative path for the user login. The user must have write permission to the directory.

You can also compress archived log files into gzip format before exporting them to external FTP servers. The compressed filename has a .gz extension in the filename. This compression feature uses less disk space than that required for noncompressed archived files on both the Content Engine and the FTP export server and also requires less bandwidth during export because of the smaller size of the files to be exported.

Restarting Export After Receiving a Permanent Error from the External FTP Server

When an FTP server returns a permanent error to the Content Engine, the archive transaction logs are no longer exported to that server. You must reenter the Content Engine transaction log export parameters for the misconfigured server to clear the error condition.

A permanent error (Permanent Negative Completion Reply, RFC 959) occurs when the FTP command to the server cannot be accepted, and the action does not take place. Permanent errors can be caused by invalid user logins, invalid user passwords, and attempts to access directories with insufficient permissions or directories that do not exist.

Exporting Transaction Logs to External SFTP Servers

You can also export transaction logs to a Secure File Transfer Protocol (SFTP) server. You must first enable the feature and configure the SFTP server parameters. The following information is required for each target SFTP server:

- SFTP server IP address or the host name

  The Content Engine translates the host name with a DNS lookup and then stores the IP address in the configuration.

- SFTP user login and user password
- Path of the directory where transferred files are written

  Use a fully qualified path or a relative path for the user login. The user must have write permission to the directory.

Secure File Transfer Protocol Access for Nonadministrative Users

In the ACNS 5.3.5 software release, the SFTP server on the Content Engine was enhanced to allow nonadministrative users (that is, a user with a nonzero UID) to use SFTP to access the Content Engine. In the ACNS 5.3.5 software release, the **sshd allow-non-admin-users** and **no sshd allow-non-admin-users** global configuration commands were added to enable and disable this new feature. By default, this feature is disabled on the Content Engine, and nonadministrative users cannot use SFTP to access the Content Engine. To enable this feature, enter the **sshd allow-non-admin-users** command on the Content Engine. After enabling this feature, you can disable it again by entering the **no sshd allow-non-admin-users** command on the Content Engine.

If this feature is enabled, the output of the **show running-config** EXEC command shows that this feature is enabled on the Content Engine.

Enabling Transaction Logging with the Content Distribution Manager GUI

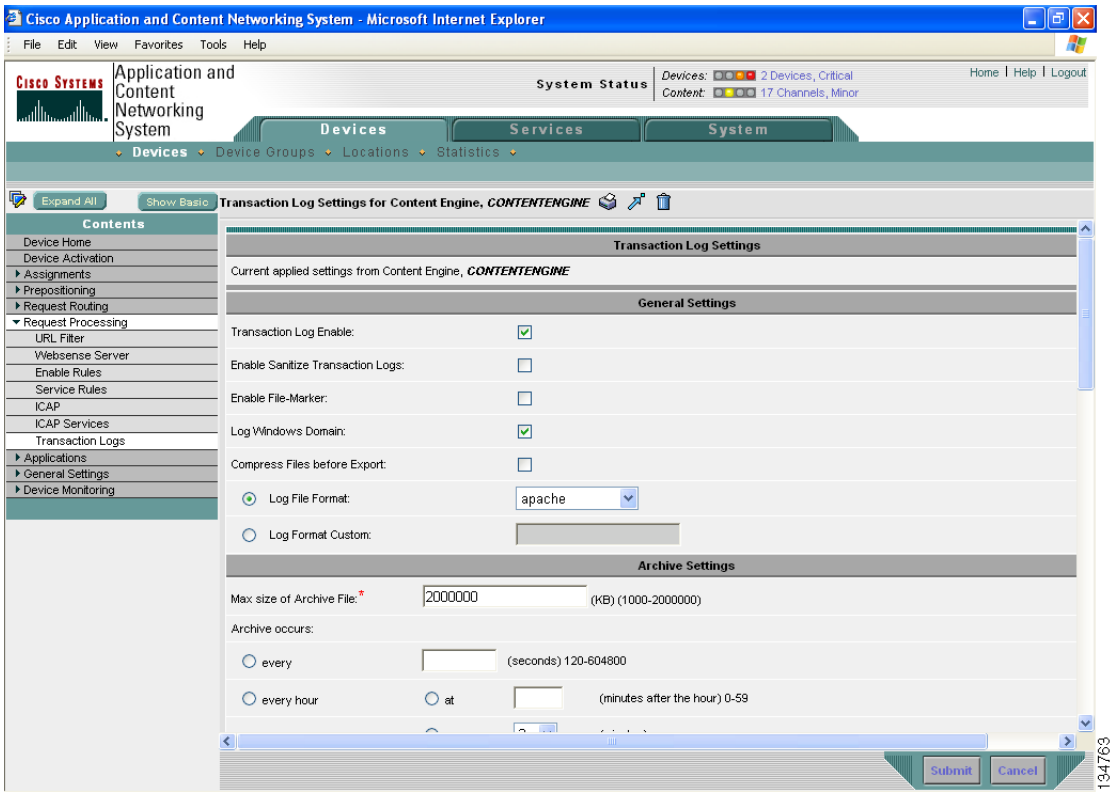
To enable transaction logging, follow these steps:

Step 1 From the Content Distribution Manager GUI, choose **Devices > Devices**.

Step 2 Click the **Edit** icon next to the name of the Content Engine that you want to configure. The Contents pane appears on the left.

From the Contents pane, choose **Request Processing > Transaction Logs**. The Transaction Logs settings window appears. (See Figure 19-1.) Table 19-4 describes the fields in this window and provides the corresponding CLI global configuration commands.

Figure 19-1 Transaction Log Settings Window—General Settings



Step 3 Under the General Settings heading, to activate transaction logging on your device, check the **Transaction Log Enable** check box.

Step 4 To disguise the IP address and username of clients, check the **Enable Sanitize Transaction Logs** check box.

Step 5 To add markers to the transaction logs that indicate where the files begin and end, check the **Enable File-Marker** check box.

Step 6 To record the Windows domain name and username in the "authenticated username" field of the transaction log, check the **Log Windows Domain** check box.

Note This option is operational if your device is configured for NT LAN Manager (NTLM) authentication and uses either the Apache-style or the Extended Squid-style transaction log format.

Step 7 To enable compression of archived log files into gzip format before exporting them to external FTP servers, check the **Compress Files before Export** check box.

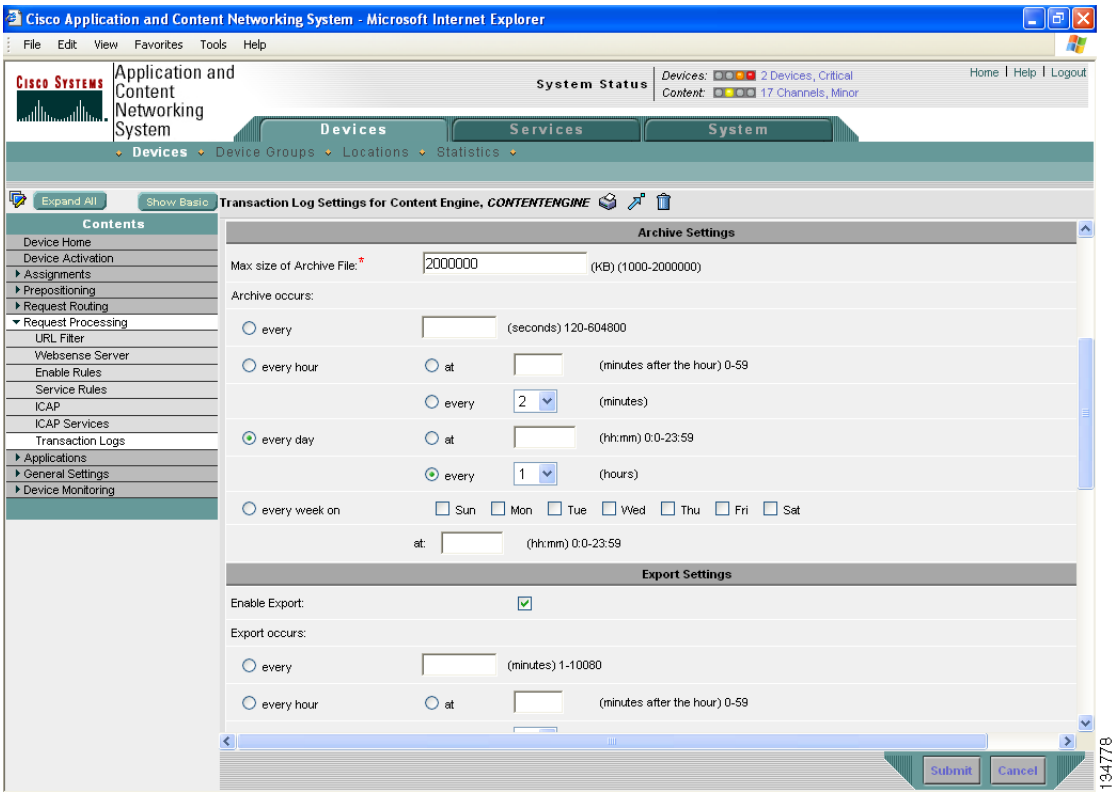
Step 8 To choose a log file format, click the **Log File Format** radio button, and choose a log file format from the drop-down list. Choose **apache**, **extended-squid**, or **squid**.

Alternatively, if you want to use a custom format for the transaction log, click the **Log Format Custom** radio button, and then enter a custom format string in the field provided. (See the "Custom Format Transaction Logging" section on page 19-2.)

Step 9 Under the Archive Settings heading (see Figure 19-2), in the Max Size of Archive File field, specify a value for the maximum size of the archive file in kilobytes. Table 19-4 describes the fields in this window and provides the corresponding CLI global configuration commands.

This value is the maximum size of the archived file to be maintained on the local disk.

Figure 19-2 Transaction Log Settings Window—Archive Settings



Step 10 To schedule the interval when the working log should be cleared by moving data into the archive log, choose a time option from among the options given in the **Archive occurs** section.

Step 11 To save the settings, click **Submit**.

Table 19-4 Transaction Log General and Archive Settings

GUI Parameter	Function	CLI Command
General Settings		
Transaction Log Enable	Enables transaction logging on the Content Engine.	**transaction-logs enable**
Enable Sanitize Transaction Logs	Disguises the IP address and username of clients.	**transaction-logs sanitize**
Enable File-Marker	Adds markers to the transaction logs to indicate where files begin and end.	**transaction-logs file-marker**
Log Windows Domain	Records the Windows domain name and username in the “authenticated username” field of the transaction log.	**transaction-logs log-windows-domain**

Table 19-4 Transaction Log General and Archive Settings (continued)

GUI Parameter	Function	CLI Command
Compress Files before Export	Enables compression of archived log files into gzip format before exporting them to external FTP servers.	**transaction-logs export compress**
Log File Format	Configures the log file format (**apache**, **extended-squid**, or **squid**).	**transaction-logs format** {**squid** | **extended-squid** | **apache**}
Log Format Custom	Configures a custom log file format.	**transaction-logs format custom** *string*
Archive Settings		
Max Size of Archive File	Maximum size (in kilobytes) of the archive file to be maintained on the local disk.	**transaction-logs archive max-file-size** *kilobytes*
Archive occurs every (interval)	Interval for the working log to be cleared by moving data into the archive log.	**transaction-logs archive interval** {*seconds* | **every-week** [**on** *weekdays* **at** *hour:minute*] | **every-day** {**at** *hour:minute* | **every** *hours*} | **every-hour** {**at** *minute* | **every** *minutes*}}

If you want to enable exporting to an FTP server, follow these steps:

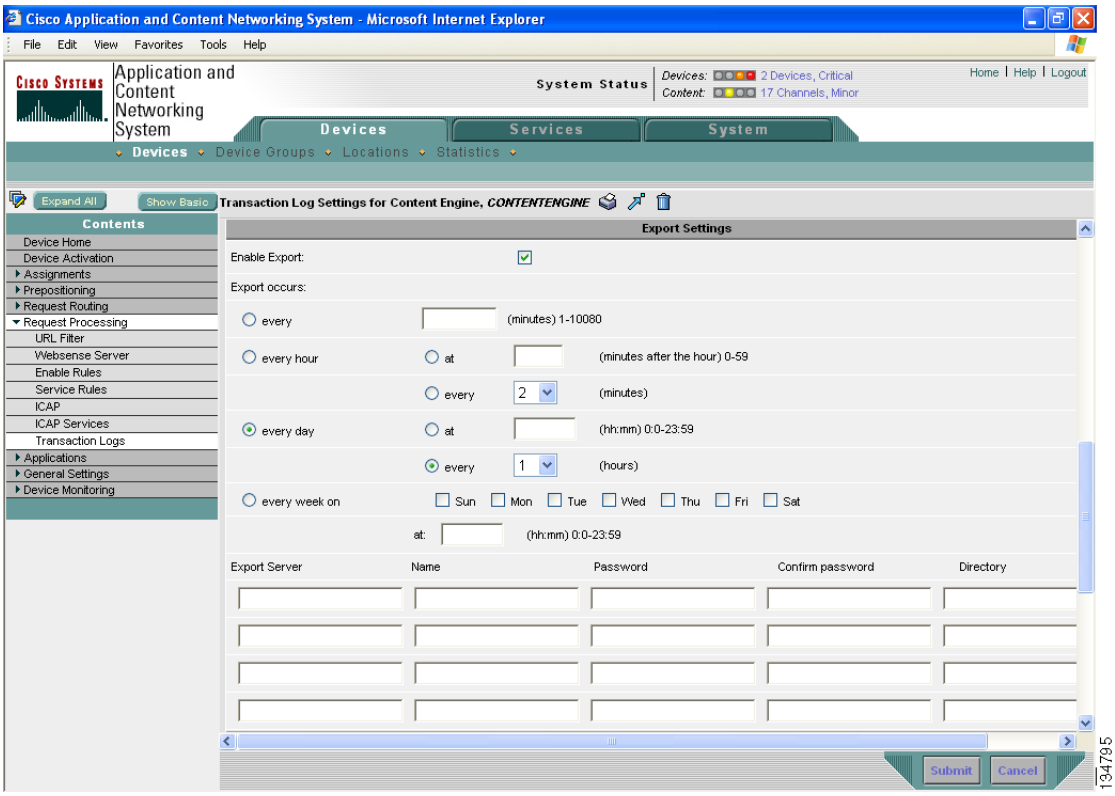
Step 1 From the Content Distribution Manager GUI, choose **Devices > Devices**.

Step 2 Click the **Edit** icon next to the name of the Content Engine that you want to configure. The Contents pane appears on the left.

Step 3 From the Contents pane, choose **Request Processing > Transaction Logs**. The Transaction Logs settings window appears.

Step 4 Under the Export Settings heading, check the **Enable Export** check box. (See Figure 19-3.) Table 19-5 describes the fields in this window and provides the corresponding CLI global configuration commands.

Figure 19-3 Transaction Log Settings Window—Export Settings



Step 5 Using the time options shown, specify the interval when the working log should be cleared by moving data into the FTP server.

Step 6 Enter an IP address or host name information for the FTP server in the Export Server field.

Note The FTP export feature can support up to four servers. Each server must be configured with a username, password, and directory that are valid for that server.

Step 7 In the Name field, enter a user ID.

Step 8 In the Password and Confirm Password fields, enter a password for the user specified in Step 7.

Step 9 In the Directory field, enter the name of a working directory that will contain the transaction logs.

Note The user specified in the Name field must have write permission to the specified directory.

Step 10 If the server chosen is a secure FTP server, check the **SFTP** check box.

Step 11 To save the settings, click **Submit**.

Table 19-5 Transaction Log Export Settings

GUI Parameter	Function	CLI Command
Enable Export	Enables transaction log export to an external FTP server.	**transaction-logs export enable**
Export occurs (interval)	Interval for the working log to be cleared by moving data to the FTP or SFTP server.	**transaction-logs export interval** {*minutes* | **every-week** [**on** *weekdays* **at** *hour:minute*] | **every-day** {**at** *hour:minute* | **every** *hours*} | **every-hour**{**at** *minute* | **every** *minutes*}}
Export Server	IP address or hostname of the FTP server.	**transaction-logs export ftp-server** {*hostname* | *servipaddrs*} *login passw directory*
Name	Name of the user.	
Password	Password for the user.	
Confirm Password	Confirms the password for the user.	
Directory	Name of a working directory that will contain the transaction logs.	
SFTP	Configures a secure FTP server.	**transaction-logs export sftp-server** {*hostname* | *servipaddrs*} *login passw directory*

Using WMT Transaction Logging

For certain companies, streaming media is a source of revenue and needs to be tracked closely. Because these companies charge their customers to stream on-demand content and live broadcasts, they must rely on logged information to track what content a particular customer viewed, how long they viewed it, and what their viewing quality was. Consequently, the accuracy and reliability of transaction logging is very important to these companies.

The Windows Media Services 9 Series provides a more robust logging model than Windows Media Services version 4.1. ACNS 5.5 software supports Windows Media Services 9 logging.

Note In ACNS software (release 5.1 and earlier), only the standard Windows Media Services version 4.1 and the extended Windows Media Services version 4.1 logging formats were supported.

In ACNS 5.5 software, the following logging formats are supported for WMT transaction logging:

- Standard Windows Media Services version 4.1
- Extended Windows Media Services version 4.1
- Standard Windows Media Services version 9.0
- Extended Windows Media Services version 9.0

The extended versions of the logging formats contain additional fields that are Content-Engine specific (For example, the CE-action field specifies a cache hit or miss, and the CE-bytes field specifies the number of bytes that were sent from the Content Engine.)

The Content Engine's transaction logging format for WMT streaming is consistent with that of the Windows Media Services and the World Wide Web Consortium (W3C)-compliant log format. A log line is written for every stream accessed by the client. The location of the log is not configurable. These logs can be exported using FTP. When transaction logging is enabled, daemons create a separate *working.log* file in /local1/logs/export for WMT transactions.

All client information in the transaction logs is sent to the origin server by default.

Log Formats Accepted by Windows Media Services 9

Windows Media Players connect to a Windows Media server using the following protocols:

- Windows Media Players earlier than Version 9.0 use HTTP/1.0 or the MMS protocol.
- Windows Media Player Version 9.0 uses HTTP/1.1 and RTSP.

Depending on the version of the Windows Media Player, logs are sent in different formats, such as text, binary, or Extensible Markup Language (XML). Table 19-6 describes the log formats accepted by Windows Media Services 9.

Table 19-6 Windows Media Services 9 Log Formats

Protocol	Player and Distributor	Log Type
HTTP/1.0	Windows Media Player earlier than Version 9.0 Content Engine (caching and proxy server) is running Windows Media Services Version 9.0 and streaming from a WMT server that is running Windows Media Services 4.1	World Wide Web Consortium (W3C) standard space-delimited text log
MMS	Windows Media Player earlier than Version 9.0	Binary structure log
HTTP/1.1	Windows Media Player Version 9.0 Distribution server is running Windows Media Services 9.0 Content Engine (caching and proxy server) is running Windows Media Services 9.0	XML structure log
RTSP	Windows Media Player Version 9.0 Distribution server is running Windows Media Services 9.0 Content Engine (caching and proxy server) is running Windows Media Services 9.0	XML structure log

Note ACNS 5.5 software supports Extensible Markup Language (XML) logging for MMS-over-HTTP and MMS-over-RTSP (RTSP over Windows Media Services 9). The posted XML log file from the Windows Media Player to the Content Engine (Windows Media server) can be parsed and saved to the normal WMT transaction logs that are stored on the Content Engine.

Configuring WMT Transaction Logging

To configure WMT transaction logging, follow these steps:

Step 1 Navigate to the Transaction Logs Settings for Content Engine window.

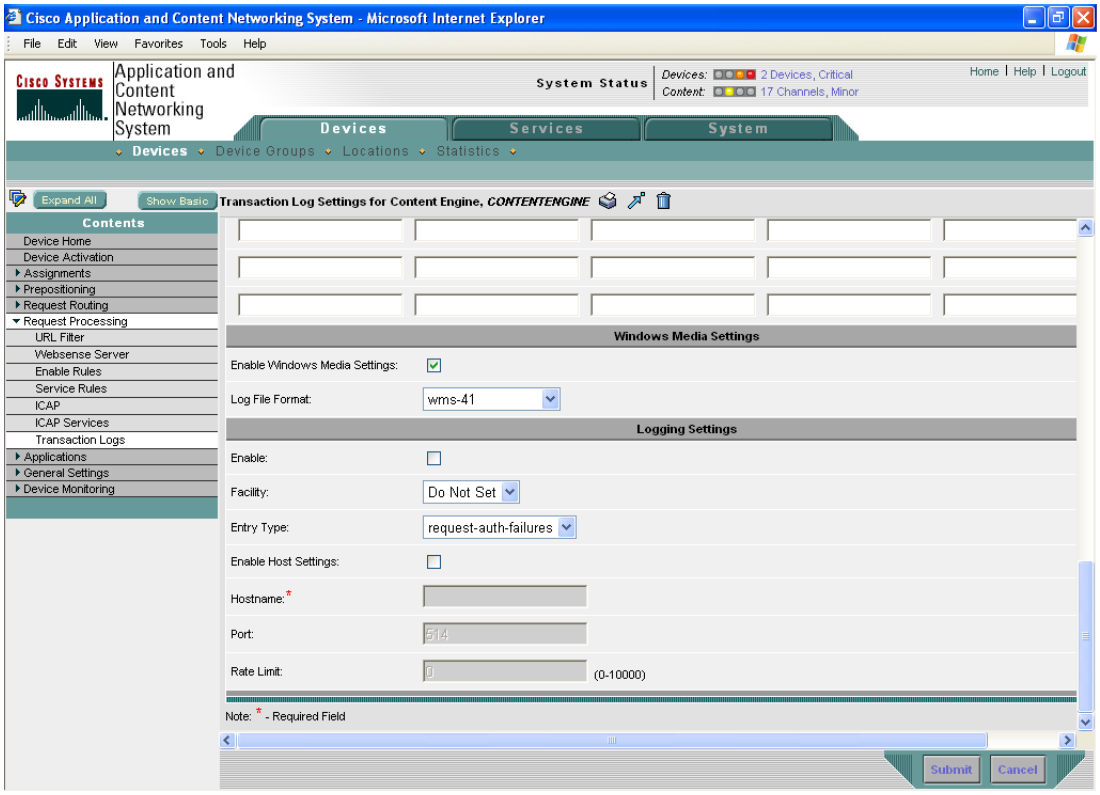
a. From the Content Distribution Manager GUI, choose **Devices > Devices**.

b. Click the **Edit** icon next to the name of the Content Engine that you want to configure. The Contents pane appears on the left.

c. From the Contents pane, choose **Request Processing > Transaction Logs**. The Transaction Log Settings window appears. (See Figure 19-1.)

Step 2 Under the WMT Settings heading (see Figure 19-4), to generate transaction logs for WMT streaming sessions, check the **Enable WMT Settings** check box.

Figure 19-4 Transaction Log Settings Window—WMT and Logging Settings



Step 3 From the **Log File Format** drop-down list, choose the logging format for WMT transaction logs. Table 19-7 describes the drop-down list options.

Table 19-7 WMT Log File Format Options

Log Format	Description
extended	Specifies the WMT extended configurations for transaction logs. Enables username logging in the WMT transaction log.
wms-41	Sets WMT to generate transaction logs in extended Windows Media Services 4.1 format. When you use this option, the Content Engine uses the standard Windows Media Services 4.1 format to generate the transaction log and includes the following three additional fields in the transaction log: • CE_action (cache hit or cache miss) • CE-bytes (number of bytes sent from the Content Engine for a cache hit) • username (username of the WMT request when NTLM, Negotiate, Digest, and basic authentication is used)
wms-90	Sets WMT to generate transaction logs in extended Windows Media Services 9 format. When you use this option, the Content Engine uses the standard Windows Media Services 9 format to generate the transaction log and includes the following three additional fields in the transaction log: • CE_action (cache hit or cache miss) • CE-bytes (number of bytes sent from the Content Engine for a cache hit) • username (username of the WMT request when NTLM, Negotiate, Digest, and basic authentication is used)
wms-41	Sets WMT to generate transaction logs in the standard Windows Media Services 4.1 format.
wms-90	Sets WMT to generate transaction logs in the standard Windows Media Services 9 format.

To configure WMT transaction logging from the CLI, use the following global configuration command:

wmt transaction-logs format {**extended** {**wms-41** | **wms-90**} | **wms-41** | **wms-90**}

Using Real-Time Transaction Logging

You can monitor transaction logs in real-time for particular errors such as authentication errors. By sending HTTP transaction log messages to a remote syslog server, you can monitor the remote syslog server for HTTP request authentication failures in real-time. This real-time transaction log feature allows you to monitor transaction logs in real-time for particular errors such as HTTP request authentication errors. The existing transaction logging to the local file system remains unchanged.

For this purpose, you must configure the Content Engine to send transaction log messages to a remote syslog server using UDP as the transport protocol. Because UDP is an unreliable transport protocol, message transport to remote syslog host is not reliable and you must monitor the syslog messages received at the remote syslog server. You can limit the rate at which the transaction logging module is allowed to send messages to the remote syslog server. The format of the syslog message is in standard syslog message format with the transaction log message as the payload of the syslog message.

Real-time transaction logging to a remote syslog server uses the standard syslog message format with the message payload as the transaction log entry. A new syslog error identifier is defined for this type of real-time transaction log message. You can configure the Content Engine to send transaction log messages in real-time to one remote syslog host. The message format of the transaction log entry to the remote syslog host is the same as in the transaction log file and prepended with Cisco's standard syslog header information.

The following is an example of the format of the real-time syslog message sent from the transaction logging module (Content Engine) to the remote syslog host:

fac-pri Apr 22 20:10:46 ce-host cache: %CE-TRNSLG-6-460012: *translog formatted msg*

The fields in the message are described as follows:

- *fac-pri* denotes the facility parameter and priority for transaction log messages encoded (as in standard syslog format) as a 32-bit decimal value between 0 and 1023 (0x0000 and 0x03FF). The least significant 3 bits denote priority (0-7) and the next least significant 7 bits denote facility (0-127).

  The facility parameter used by the transaction logging module when a real-time transaction log message is logged to the remote syslog host is "user". The same facility is sent to the remote syslog host unless you configure a different facility parameter for transaction logging. The priority field is always set to LOG_INFO for real-time transaction log messages.

  In the above example, the default value of *fac-pri* is 14 (0x000E) where facility = user (LOG_USER (1)) and priority = LOG_INFO (6).
- The next field in the message is the date, which follows the format as shown in the above example.
- *ce-host* is the hostname or IP of the Content Engine that is sending the message.
- *cache* is the name of the process on the Content Engine that is sending the message.
- %CE-TRNSLG-6-460012 is the Cisco standard formatted syslog header on the Content Engine for a real-time transaction log message. This identifier indicates a priority level of 6, which denotes informational messages.

Note The Content Engine system syslog messages report communication errors with the remote syslog host that is configured for transaction logging. These syslog messages are in the error message range: %CE-TRNSLG-6-460013 to %CE-TRNSLG-3-460016. Note that the last error message (%CE-TRNSLG-3-460016), shows level "3" (for error-level messages) instead of "6" (for information-level messages). Information-level messages are reported when messages are dropped due to rate limiting and the number of dropped messages are reported. For more information about these syslog messages, refer to the ACNS Syslog Error Book.

- *translog formatted msg* is the transaction log message as it appears in the transaction log file.

Note The total length of the real-time syslog message is 1024 characters. If the actual transaction log entry exceeds this limit, it is truncated.

To include the username and domain name in the transaction log, check the **Log Windows Domain** check box or use the **transaction-logs log-windows-domain** global configuration command.

When the remote syslog server logs this message to a file, the format appears as follows:

Apr 22 20:10:46 ce-host cache: %CE-TRNSLG-6-460012: *translog formatted msg*

where ce-host is the host name of the Content Engine that sent the real-time transaction log message to the remote syslog server.

The configuration of host settings for transaction logs is identical to the configuration settings for syslog messages except that you need not specify the priority level of the message for real time transaction logs. All messages are associated with the priority level of 6 (LOG_INFO). You are not required to filter messages based on priority levels.

Configuring Real-Time Transaction Logging

To configure real-time transaction logging, follow these steps:

Step 1 Navigate to the Transaction Logs Settings for Content Engine window.

a. From the Content Distribution Manager GUI, choose **Devices > Devices**.

b. Click the **Edit** icon next to the name of the Content Engine that you want to configure. The Contents pane appears on the left.

c. From the Contents pane, choose **Request Processing > Transaction Logs**. The Transaction Logs settings window appears. (See Figure 19-1.)

Step 2 Under the Logging Settings section (see Figure 19-4), to enable real-time transaction logging, check the **Enable** check box. You can retain the logging host configuration for transaction logs even if you temporarily disable real-time transaction logging by unchecking the check box. This new logging option applies only to the cache's HTTP transaction log entries. The real-time transaction logging feature is disabled by default.

Note You must check the **Transaction Log Enable** check box in the General settings section before you check the **Enable** check box. Otherwise, the second setting does not apply unless you enable transaction logging for the entire Content Engine.

Step 3 From the Facility drop-down list, choose the appropriate transaction log facility.

This drop-down list is set to an initial value of *Do not set*. This setting denotes that the facility sent to the syslog host will be the facility on the local host that is sending the syslog message. For instance, in the case of the transaction logging module that sends the real-time transaction log message, the facility is the "user" facility.

Step 4 From the Entry Type drop-down list, choose the type of transactions that you want to log from the Content Engine to the remote syslog host. Choose **request-auth-failures** to send only those transactions associated with HTTP request authentication failures to the remote syslog host. Choose **all** to send all of the transaction messages to the remote syslog host. See the "About Specifying the Transaction Log Entry Type when Logging to a Remote Syslog Host" section on page 19-21.

Step 5 To enable sending of transaction log files to a remote syslog host, check the **Enable Host Settings** check box.

Step 6 In the Hostname field, enter a host name or an IP address of the remote syslog server to which transaction logs must be sent. No remote syslog server is specified by default.

Step 7 In the Port field, specify the destination port on the remote syslog host to which the Content Engine should send the message. The default port number is 514. This port is a well-known port for system logging.

Step 8 In the Rate Limit field, specify the number of messages that are allowed to be sent to the remote syslog host per second. To limit bandwidth and other resource consumption, messages to the remote syslog host can be rate limited. If this limit is exceeded, the specified remote syslog host drops the messages. There is no default rate limit (rate-limit is set to 0), and by default all syslog messages are sent to all of the configured syslog hosts. The range is 1 to 10,000 messages per second.

Step 9 To save the settings, click **Submit**.

A "Click Submit to Save" message appears in red next to the Current Settings line when there are pending changes to be saved after you have applied default or device group settings. You can also revert to the previously configured settings by clicking **Reset**. The **Reset** button is visible only when you have applied default or group settings to change the current device settings but have not yet submitted the changes.

If you try to leave this window without saving the modified settings, a warning dialog box prompts you to submit the changes. This dialog box only appears if you are using the Internet Explorer browser.

About Specifying the Transaction Log Entry Type when Logging to a Remote Syslog Host

You can configure the Content Engine to send only the transactions associated with HTTP request authentication failures, or to send all of the transactions.

Typically, organizations are interested in only HTTP request authentication failures for security purposes. By monitoring these types of authentication failures in real time, it enables organizations to identify which end users failed to be authenticated through the Content Engine.

Only the authentication failure transaction that is associated with an end user who is attempting to contact the authentication server is logged. The "pending" transactions that are waiting for a response from the transaction that contacted the authentication servers are not logged. This approach provides you with the information needed to determine which user fails to authenticate with the Content Engine and minimizes the traffic to the syslog host. To track which users failed to authenticate, you must configure a transaction log format that logs the username by configuring either Extended Squid-style format or the custom log format with the custom format token %u . For more information about specifying the format of the transaction log, see the "Understanding Transaction Log Formats" section on page 19-1 and the "Custom Format Transaction Logging" section on page 19-2.

When you check the **Enable** check box to enable real-time transaction logging (or specify the **transaction-logs logging enable** global configuration command), the logging of only HTTP request authentication failures is the default. If you want to change this default and log all transactions, then you must choose **all** from the Entry Type drop-down list (or enter the **transaction-log logging entry-type all** global configuration command on the Content Engine). However, if you log all transactions, there may be a significant UDP drop rate if your syslog host cannot handle the rate of incoming traffic.