



Configuring Transaction Logging

This chapter explains transaction logging and describes the different transaction logging formats available on the Content Engine. This chapter contains the following sections:

- [Transaction Logging Overview, page 14-1](#)
- [Squid-Style Transaction Logging, page 14-2](#)
- [Extended Squid-Style Transaction Logging, page 14-3](#)
- [Apache-Style Transaction Logging, page 14-3](#)
- [Transaction Logging and NTLM Authentication, page 14-4](#)
- [Sanitized Transaction Logs, page 14-4](#)
- [Exporting Log Files, page 14-5](#)

Transaction Logging Overview

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.

High-performance caching presents additional challenges other than how to quickly retrieve objects from storage, memory, or the web. Administrators of caches are often interested in what requests have been made of the cache and what the results of these requests were. This information is then used for such applications as:

- Problem identification and solving
- Load monitoring
- Billing
- Statistical analysis
- Security problems
- Cost analysis and provisioning

In ACNS 4.2 software, the user can choose between Squid, Extended Squid, and Apache log formats.

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://www.squid-cache.org/doc/FAQ/FAQ-6.html>

The Squid log file format is:

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 -
```

Squid logs are a valuable source of information about cache workloads and performance. The logs record not only access information but also system configuration errors and resource consumption, such as memory and disk space.

This example enables transaction logging in Squid-style format.

```
ContentEngine(config)# transaction-logs format squid
ContentEngine(config)#
```

Table 14-1 lists the fields associated with the Squid-style format.



Note

Many public tools are available that can convert a Squid-style transaction log into reports. Visit the following website, <http://www.squid-cache.org/Scripts> for listings of such tools.

Table 14-1 Squid-Style Format Description

Field	Description
Time	UNIX time stamp as Coordinated Universal Time (UTC) seconds with a millisecond resolution.
Elapsed	Length of time in milliseconds that the cache was busy with the transaction. Note Entries are logged after the reply has been sent, not during the lifetime of the transaction.
Remote Host	IP address of the requesting instance.
Code/Status	Two entries separated by a slash. The first entry contains information on the result of the transaction: the kind of request, how it was satisfied, or in what way it failed. The second entry contains the HTTP result codes.
Bytes	Amount of data delivered to the client. This does not constitute the net object size, because headers are also counted. Also, failed requests may deliver an error page, the size of which is also logged here.
Method	Request method to obtain an object for example, GET.
URL	URL requested.

Table 14-1 Squid-Style Format Description (continued)

Field	Description
Rfc931	Contains the authentication server's identification or lookup names of the requesting client. This field will always be a "-" (dash).
Peerstatus/Peerhost	Two entries separated by a slash. The first entry represents a code that explains how the request was handled, for example, by forwarding it to a peer, or returning the request to the source. The second entry contains the name of the host from which the object was requested. This host may be the origin site, a parent, or any other peer. Also note that the host name may be numerical.
Type	Content type of the object as seen in the HTTP reply header. In ACNS 4.2 software, this field will always contain a "-" (dash).

Extended Squid-Style Transaction Logging

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 (Table 14-1) 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 -
```

Use the **transaction-logs format extended-squid** command to enable transaction logging in Extended Squid format.

This example shows how to enable transaction logging in Extended Squid format.

```
ContentEngine(config)# transaction-logs format extended-squid
ContentEngine(config)#
```

Apache-Style Transaction Logging

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

The Apache-style log file format is:

```
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
```

Table 14-2 lists the fields associated with the Apache CLF format.

Table 14-2 Apache Common Log File Format Descriptions

Field	Description
Remotehost	Remote host name or IP address.
Rfc931	Contains the authentication server's identification or lookup names of the requesting client. This field will always contain a "-" (dash).
Authuser	Username that the user entered for authentication purposes. This will be a "-" (dash) if no user information is available.
Date	Date and time of request.
Request	First line of the request.
Status	HTTP status code, for example, 200.
Bytes	Content length of the document transferred.

This example shows how to enable transaction logging in Apache style format.

```
ContentEngine(config)# transaction-logs format apache
ContentEngine(config)#
```

Transaction Logging and NTLM Authentication

If your device is configured for NT LAN Manager (NTLM) authentication and uses Apache-style as well as Extended Squid-style format, the **transaction-logs log-windows-domain** command records 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 "-" (dash) is recorded in the field.

Use the **no transaction-logs log-windows-domain** command to negate logging NTLM parameters in Apache-style and Extended-style formats.

Sanitized Transaction Logs

Use the **transaction-logs sanitized** command to disguise the IP address and usernames of clients in the transaction log file. The default is 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. The **no** form disables the sanitize feature.

This examples shows how to enable the sanitize feature in transaction logging.

```
ContentEngine(config)# transaction-logs sanitize
ContentEngine(config)#
```

Exporting Log Files

In order to facilitate the postprocessing of cache log files, it is possible to export transaction logs to an external host.

This feature allows for 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 of:

```
<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 MMS type 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 the feature and then configure the FTP server parameters. Use the **transaction-logs export enable** command to enable exporting of transaction logs to external FTP servers. You can then use the **transaction-logs export ftp-server** option to enter FTP server parameters. This option 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.

Use the **transaction-logs export compress** command to compress archived log files into gzip format prior to exporting them to external FTP servers. The compressed filename has a .gz extension in the filename. This feature uses less disk space for the archived files on both the Content Engine and the FTP export server and also requires less bandwidth during export.

In this example, two FTP servers are configured.

```
ContentEngine(config)# transaction-logs export ftp-server 10.1.1.1 mylogin mypasswd
/ftpdirectory
ContentEngine(config)# transaction-logs export ftp-server myhostname mylogin mypasswd
/ftpdirectory
```

To delete an FTP server, use the **no** form of the command.

```
ContentEngine(config)# no transaction-logs export ftp-server 10.1.1.1
ContentEngine(config)# no transaction-logs export ftp-server myhostname
```

Use the **no** form of the **transaction-logs export enable** command to disable the entire transaction log export feature while retaining the rest of the configuration.

```
ContentEngine(config)# no transaction-logs export enable
```

To change a username, password, or directory, reenter the entire line.

```
ContentEngine(config)# transaction-logs export ftp-server 10.1.1.1 mynewname mynewpass
/newftpdirectory
```

The **show transaction-logging** command displays information on the configuration of transaction logging in the Content Engine. Note that transaction log file information is displayed for HTTP and WMT Microsoft Media Server (MMS) caching proxy transactions.

```
ContentEngine# show transaction-logging
Transaction log configuration:
-----
Logging is enabled.
Logging of ecdn internal communication is disabled.
End user identity is hidden. (sanitized)
File markers are disabled.
Archive interval: every-day every hour.
Maximum size of archive file: 2000000 KB
Log File format is extended-squid.

Exporting files to ftp servers is enabled.
File compression is disabled.
Export interval: every-day at 11:45 local time

ftp-server      username      directory
10.1.1.1        mylogin      /ftpdirectory
10.2.2.2        mylogin      /ftpdirectory
HTTP Caching Proxy Transaction Log File Info
  Working Log file - size : 83
                    age: 502845
  Archive Log file - celog_10.1.1.21_20020107_150300.txt      size: 1075
  Archive Log file - celog_10.1.1.21_20020117_150300.txt      size: 1199746
  Archive Log file - celog_10.1.1.21_20020118_000000.txt      size: 137583
  Archive Log file - celog_10.1.1.21_20020118_150300.txt      size: 12667
  Archive Log file - celog_10.1.1.21_20020123_150300.txt      size: 298
WMT MMS Caching Proxy/Server Transaction Log File Info
  Working Log file - size : 541
                    age: 54117
  Archive Log file - mms_export_10.1.1.21_20020107_225942      size: 541
  Archive Log file - mms_export_10.1.1.21_20020107_232156      size: 938
  Archive Log file - mms_export_10.1.1.21_20020117_193239      size: 541
  Archive Log file - mms_export_10.1.1.21_20020122_224556      size: 1993
  Archive Log file - mms_export_10.1.1.21_20020124_150334      size: 541
  Archive Log file - mms_export_10.1.1.21_20020131_025505      size: 541
ContentEngine#
```



Note For security reasons, passwords are never displayed.

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. The **show statistics transaction-logs** command displays the current state of transaction log export readiness.

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.

In the following example, an invalid user login parameter was included in the **transaction-logs export ftp-server** command. The **show statistics transaction-logs** command shows that the Content Engine failed to export archive files.

```
ContentEngine# show statistics transaction-logs
Transaction Log Export Statistics:

Server:172.16.10.5
  Initial Attempts:1
  Initial Successes:0
  Initial Open Failures:0
  Initial Put Failures:0
  Retry Attempts:0
  Retry Successes:0
  Retry Open Failures:0
  Retry Put Failures:0
  Authentication Failures:1
  Invalid Server Directory Failures:0
```

To restart the export of archive transaction logs, you must reenter the **transaction-logs export ftp-server** parameters.

```
ContentEngine(config)# transaction-logs export ftp-server 172.16.10.5 goodlogin pass
/ftpdirectory
```

