



Configuring ICAP on Standalone Content Engines

This chapter describes how to configure the Internet Content Adaptation Protocol (ICAP) on standalone Content Engines. ACNS 5.2 software supports ICAP for HTTP and FTP-over-HTTP requests. Support for native FTP requests is not supported.

This chapter includes the following topics:

- [Overview of ICAP, page 11-1](#)
- [Scenario of Configuring ICAP Services on a Content Engine, page 11-4](#)
- [Overview of Configuring ICAP for Standalone Content Engines, page 11-5](#)
- [Configuring ICAP Settings for Standalone Content Engines, page 11-5](#)
- [Configuring ICAP Services on Standalone Content Engines, page 11-6](#)
- [Configuring an ICAP Server for Standalone Content Engines, page 11-8](#)
- [Configuring Logging of ICAP Exchanges, page 11-9](#)
- [Displaying Information About an ICAP Configuration, page 11-9](#)
- [Displaying Statistics for ICAP Services, page 11-10](#)



Note

For complete syntax and usage information for the CLI commands used in this chapter, refer to the *Cisco ACNS Software Command Reference, Release 5.2* publication.

For information about how to configure ICAP for Content Engines that are registered with a Content Distribution Manager, refer to the *Cisco ACNS Software Configuration Guide for Centrally Managed Deployments, Release 5.2*.

Overview of ICAP

ICAP is an open standards protocol for content adaptation, typically at the network edge. Content adaptation includes virus scanning, content translation, content filtering, content insertion, and other ways of improving the value of content to end users. ICAP specifies how a Content Engine, acting as an HTTP proxy server, can communicate with an external device that is acting as an ICAP server, which filters and adapts the requested content.

ICAP provides two content-processing modes for HTTP services. These modes define the transactions that can occur between a Content Engine acting as an ICAP client and an ICAP server. The two modes are:

- Request modification (**reqmod**)—Allows modification of requests as they are sent from the Content Engine to the ICAP server on their way to the origin server. The ICAP server can modify these requests depending on the services requested.
- Response modification (**respmod**)—Allows modification of requests after they return from the origin server. The ICAP server only acts on requested objects, after they return from the origin server.

About ICAP Services

An ICAP service is a collection of attributes that define the service and one or more ICAP servers that provide the ICAP services. You can configure a maximum of ten ICAP services per Content Engine, with an upper limit of five ICAP servers per ICAP service. Also, you can choose to apply ICAP services on all HTTP requests processed by the Content Engine or apply ICAP processing only to requests that match the Rules Template.



Tip

To set the type of load balancing to use among a cluster of ICAP servers, use the **icap service load balancing** global configuration command.

About ICAP Services and Vectoring Points

The point at which ICAP services are applied to content is called the *vectoring point*, specified using the **vector-point** option. The following three vectoring points are supported:

- Client request vectoring point (**reqmod-postcache**)—The ICAP server performs one of the following actions in response to the client request:
 - Terminates the connection
 - Sends a modified error response
 - Searches the cache using the URL in the request
 - Searches the cache using a modified URL
 - Modifies the request header or request body in the case of a cache miss
- Cache miss vectoring point (**reqmod-precache**)—The ICAP server performs one of the following actions before forwarding the request to the origin server:
 - Terminates the connection
 - Sends a modified error response
 - Sends the request to the origin server using the original URL
 - Sends the request to the origin server using an alternative URL
 - Modifies the request header or request body

- Server response vectoring point (**respmo-d-precache**)—The ICAP server performs one of the following actions after receiving the response from the origin server:
 - Returns the response to the client
 - Modifies the request header or request body
 - Caches the response using the original URL
 - Caches the response using an alternative URL

**Note**

Different ICAP services assigned to the same vectoring point can use different load-balancing options.

The following commands show a typical configuration for a virus-scanning service that requires processing on two vectoring points: **reqmod-precache** and **respmo-d-precache**:

```
ContentEngine(config)# icap apply all
ContentEngine(config)# icap service trend-reqmod
ContentEngine(config-icap-service)# enable
ContentEngine(config-icap-service)# vector-point reqmod-precache
ContentEngine(config-icap-service)# server icap://172.19.227.150/REQ-Service
ContentEngine# exit
ContentEngine# icap service trend-respmo
ContentEngine(config-icap-service)# enable
ContentEngine(config-icap-service)# vector-point respmod-precache
ContentEngine(config-icap-service)# server icap://172.19.227.150/interscan
ContentEngine# exit
```

If an ICAP vendor supports the same service name for more than one vectoring point, you can configure a single service and add the supported vectoring points, as in the following example:

```
ContentEngine(config)# icap service myicap-service
ContentEngine(config-icap-service)# enable
ContentEngine(config-icap-service)# vector-point reqmod-precache
ContentEngine(config-icap-service)# vector-point respmod-precache
ContentEngine(config-icap-service)# server icap://172.19.227.150/icap-service-name
ContentEngine(config-icap-service)# exit
ContentEngine(config)#
```

ACNS Software and ICAP Services Interoperability Notes

This section provides information about using ICAP processing services with the ACNS software.

ICAP Vendors Supported

The following is a complete list of the ICAP vendors that have been certified to interoperate with the Content Engine:

- TrendMicro for reqmod and respmod
- Symantec for respmod

Maximum File Size Supported

The maximum file size that is supported in the ACNS software in pass-through mode is 2 GB. Files that exceed this size limit are not supported for ICAP processing.

Scenario of Configuring ICAP Services on a Content Engine

The following is a sample of how ICAP services might be defined and enabled on a standalone Content Engine:

1. Use the **icap apply {all | rules-template}** global configuration command to specify which ICAP services should be performed on which requests that are received by the Content Engine. For example,

- Use the **icap apply rules-template** global configuration command to instruct the Content Engine to run only the ICAP services that match the rules action **use-icap-service**.

```
ContentEngine(config)# icap apply rules-template
```

- Alternatively, you could use the **icap apply all** global configuration command to instruct the Content Engine to run all of the ICAP services on all of the HTTP requests that it receives.

2. Use the **icap logging enable** global configuration command to turn on ICAP-related transaction logging, which is available in the local/logs/icap/ directory.

```
ContentEngine(config)# icap logging enable
```

3. Use the **icap service service-id** global configuration command to configure and enable various ICAP services on this Content Engine.

```
ContentEngine(config)# icap service trend-reqmod
ContentEngine(config-icap-service)# enable
ContentEngine(config-icap-service)# vector-point reqmod-precache
ContentEngine(config-icap-service)# server icap//172.19.227.150/REQ-Service
ContentEngine# exit
ContentEngine(config)# icap service trend-respmod
ContentEngine(config-icap-service)# enable
ContentEngine(config-icap-service)# vector-point respmod-precache
ContentEngine(config-icap-service)# server icap//172.19.227.150/interscan
ContentEngine# exit
```

4. Use the **rule** global configuration command to define the ICAP service rules for this Content Engine.

For instance, in the following example, certain traffic (such as intranet domain traffic or other trusted traffic) is intentionally prevented from going through ICAP processing:

```
ContentEngine(config)# rule enable
ContentEngine(config)# rule action use-icap-service trend-reqmod pattern-list 1
protocol all
ContentEngine(config)# rule action use-icap-service trend-respmod pattern-list 1
protocol all
ContentEngine(config)# rule pattern-list 1 domain !(*.cisco\.com|*.datek\.com)
```



Tip

By default, requests from streaming media clients are bypassed for ICAP processing. You can, however, use the **icap bypass streaming-media** global configuration command to turn on the bypass feature for streaming media if necessary.



Note

For more information about ICAP services, see the [“About ICAP Services and Vectoring Points” section on page 11-2](#).

Overview of Configuring ICAP for Standalone Content Engines

To configure ICAP on a standalone Content Engine, you must use the Content Engine CLI to complete the following tasks:

1. Configure ICAP settings on the Content Engine.

For more information on this topic, see the next section, [Configuring ICAP Settings for Standalone Content Engines](#).”

2. Configure ICAP services on the Content Engine.

To configure ICAP services, use the **icap service** global configuration command, as described in the [“Configuring ICAP Services on Standalone Content Engines”](#) section on page 11-6.

3. Configure an ICAP server for the Content Engine.

For more information on this topic, see the [“Configuring an ICAP Server for Standalone Content Engines”](#) section on page 11-8.

Configuring ICAP Settings for Standalone Content Engines

The Content Engine CLI must be used to configure ICAP settings on a standalone Content Engine. (The Content Engine GUI does not currently support the configuration of ICAP settings.)

When using the **icap** global configuration command to set ICAP parameters on a standalone Content Engine, keep the following important points in mind.

- Use the **icap append-x-headers** global configuration command to specify the ICAP extension headers that are passed to the ICAP server during the session negotiation between the Content Engine and the ICAP server.
- In ACNS software, Release 5.1 and later, you can configure the Content Engine to append the client and server IP address headers to the request that is passed to the ICAP server. This capability allows you to use your ICAP server to perform URL filtering based on the client IP address and server IP address. To enable this capability, you must use the **icap append-x-headers x-client-ip** and **icap append-x-headers x-server-ip** command options.
- In ACNS 5.2 software, the ability to configure the Content Engine to append the username and group name headers to the request that is passed to the ICAP server was added. This capability allows you to use your ICAP server to perform URL filtering based on username and group name. To support this capability, two command options were added to the **icap append-x-headers** global configuration command:
 - The **x-authenticated-user** option that allows the username information to be passed to the ICAP server for global services. This option is disabled by default. When this option is enabled (the **icap append-x-headers x-authenticated-user** option), the x-authenticated-user information is inserted into the ICAP request to the ICAP server.
 - The **x-authenticated-groups** option that allows the group name information to be passed to the ICAP server for global services. This option is disabled by default. When this option is enabled (the **icap append-x-headers x-authenticated-groups** option), the x-authenticated-groups information is inserted into the ICAP request to the ICAP server.

The currently supported authentication schemes include LDAP, NTLM, RADIUS, and TACACS+.

- Requests from streaming media clients are by default bypassed for ICAP processing.

Use the **icap bypass streaming-media** command to force strict rechecking of the cached content every time the IStag changes. The IStag is a field in the HTTP response header that allows ICAP servers to send a service-specific cookie to an ICAP client, representing the current state of the service. The IStag may change as a result of an update to the server version, to a virus-pattern-file, or to the policy.

Table 11-1 describes the **icap** global configuration command parameters.

Table 11-1 Parameters for icap Command

Parameter	Description
append-x-headers	Appends x-headers during ICAP protocol handshake. Disabled by default. Can have multiple entries for various x-headers to be appended.
x-client-ip	Appends x-client-IP headers to the request that is sent to the ICAP server. Disabled by default.
x-server-ip	Appends x-server-IP headers to the request that is sent to the ICAP server. Disabled by default.
x-authenticated-user	Appends x-authenticated-user headers to the request that is sent to the ICAP server. Disabled by default.
x-authenticated-groups	Appends x-authenticated-groups headers to the request that is sent to the ICAP server. Disabled by default.
apply	Enables ICAP processing for HTTP requests.
all	Enables ICAP processing for all HTTP requests.
rules-template	Enables ICAP processing for HTTP requests that match the Rules Template using the rule action use-icap-service global configuration command.
bypass	Enables bypassing of certain requested content.
streaming-media	Enables bypassing of streaming media.
logging	Specifies logging-related options when ICAP services are used.
enable	Enables logging when ICAP services are used.
format	Specifies the logging format.
<i>custom</i>	Specifies a customized format for logging.
<i>word</i>	Specifies the word used to identify the customized logging .
standard	Specifies the standard format for logging.

Configuring ICAP Services on Standalone Content Engines

In ACNS software, Release 5.1 or later, three vectoring points are supported to enable content adaptation, as described earlier in the [“About ICAP Services and Vectoring Points”](#) section on page 11-2.

ICAP servers configured at various vectoring points may become overloaded with HTTP requests, especially the request modification precache vectoring points because all requests pass through this point. Therefore, a cluster of ICAP servers (a load-balanced collection of ICAP servers) is made available for configuration. At a particular vectoring point, you can choose to load balance requests among the ICAP cluster of servers based on various parameters such as weighted load, client IP and server IP address-based hash, or round-robin format.

More than one ICAP service can be associated with a vectoring point. An ICAP service configured at a vectoring point can have only one load-balancing scheme, irrespective of the number of servers. However, multiple ICAP services configured at one or all of the vectoring points can have different load-balancing schemes.

**Tip**

If you click the **Aggregate Settings** radio button the ICAP Services for Content Engine window, the ICAP services that have been previously configured for device groups to which the Content Engine belongs cannot be modified or deleted. In other words, you can only view the ICAP services created for the device groups.

Use the **icap service** global configuration command to configure ICAP services for a standalone Content Engine, as follows:

```
icap service service-id { enable | error-handling [bypass | return-error] | load-balancing
[client-ip-hash | round-robin | server-ip-hash | weighted-load] | server url [max-connections
connection-number [weight percentage] | weight percentage [max-connections connection-number]] |
vector-point [reqmod-postcache | reqmod-precache | respmo_pre-cache] order order-number }
```

Use the **icap service** *service-id* command to enter ICAP configuration mode and to configure a specific ICAP service.

Replace *service-id* with a name of your choice for the current ICAP service. When you enter the **icap service** command and provide a name for the ICAP service, the system displays the ICAP service configuration prompt:

```
ContentEngine(config-icap-service)#
```

Within ICAP service configuration mode, all commands that you enter apply to the current ICAP service.

[Table 11-2](#) describes the **icap service** command parameters for configuring an ICAP service on a standalone Content Engine.

Table 11-2 Parameters for the icap service Command

Parameter	Description
<i>service-id</i>	Specifies a name of your choice for the current ICAP service.
enable	Enables ICAP services.
error-handling	Specifies error-handling options when an ICAP service is used.
bypass	Bypasses this service when an error occurs with this service.
return-error	Returns an error message to the client and ends the request.
load-balancing	Specifies a load-balancing option for this service. See Table 11-3 for a list of these load-balancing options.
client-ip-hash	Allows for load-balancing among ICAP servers using the client IP address.
round-robin	Allows for round-robin load balancing among ICAP servers.
server-ip-hash	Allows for load balancing using the ICAP server IP address.
weighted-load	Allows for load balancing using a weight scheme that specifies weight on a server basis.

[Table 11-3](#) describes the **icap service** load-balancing options.

Table 11-3 *icap service Load-Balancing Options*

Load-Balancing Type	Description
Client IP hash	Uses a hash-based algorithm based on the client IP address for load balancing the ICAP servers in the cluster.
Round-robin	Uses the round-robin method in which ICAP servers take turns processing HTTP requests.
Server IP hash	Uses a hash-based algorithm based on the server IP address for load balancing among the ICAP servers in the cluster.
Weighted	Uses a farm of ICAP servers with different load capacities.

Configuring an ICAP Server for Standalone Content Engines

ICAP servers process HTTP requests from clients based on the ICAP services configured at various vectoring points. ICAP servers perform content adaptation such as request or response modification and filtering of requests or responses at the configured vectoring points while processing HTTP requests.

You can configure the maximum number of connections and the weight that can be handled by an ICAP server in a cluster of servers. The weight parameter represents the percentage of load that can be redirected to the ICAP server. An ICAP server with a weight of 40 denotes that this server handles 40 percent of the load. If the total weight of all ICAP servers in a load-balanced cluster exceeds 100, the percentage of load for each ICAP server is recalculated as a percentage measure represented by the weight parameters.

To configure an ICAP server for a configured ICAP service on a standalone Content Engine, use the **icap service server** global configuration command, as follows:

Table 11-4 describes the **icap service server** command parameters for configuring an ICAP server for a standalone Content Engine.

Table 11-4 *Parameters for the icap service server Command*

Parameter	Description
server	Enables the ICAP server to be used for ICAP services using a URL.
<i>url</i>	URL based in the format <code>icap://ICAPserverIPAddress/servicename</code> .
keepalive-interval	(Optional). Specifies the keepalive interval. (This option was added in ACNS 5.2 software.)
<i>keepalive-interval</i>	Keepalive interval in seconds. The default is 60 seconds. Valid values are from 1 to 3600 seconds.
max-connections	(Optional). Specifies the maximum number of connections to a particular ICAP server.
<i>connection-number</i>	Maximum number of connections. (The maximum is 5000.)
weight	(Optional) Sets weight percentage for load balancing if the weighted-load load-balancing scheme is used.
<i>percentage</i>	Percentage of the load that can be redirected to this ICAP server (0–100). This parameter must not be defined if you chose the weighted-load load-balancing scheme.

Table 11-4 Parameters for the `icap service server` Command (continued)

vector-point	Specifies vectoring point or content-processing mode to be used for the specified ICAP service.
reqmod-postcache	Allows modification of requests sent by the ICAP server and cached by the Content Engine.
reqmod-precache	Allows modification of requests as they are sent from the Content Engine to the ICAP server on their way to the origin server.
respmod-precache	Allows modification of requests after they return from the origin server.
order	Specifies the order for the content-processing mode used.
<i>order-number</i>	Order in which the content-processing mode handles requests to the Content Engine.

Configuring Logging of ICAP Exchanges

Use the `icap logging` global configuration command to configure and enable transaction logging for ICAP exchanges between the external ICAP servers and standalone Content Engines.

`icap logging {enable | format [custom word | standard]}`

Specify the format of the transaction log (custom or standard).

- If you wish to create transaction logs in ICAP's standard logging format, choose **standard**.
- Choose **custom** if you wish to log additional fields not included in the standard format.



Note

Customized format for transaction logging is not supported in ACNS software, Release 5.1 or 5.2. Only standard transaction log format is available for ICAP services configured on a Content Engine.

Displaying Information About an ICAP Configuration

To display the current ICAP configuration for standalone Content Engines, use the `show icap` EXEC command. The command output shows the status of the enabled ICAP features, the service definitions, a list of vectoring points, and an ordered list of ICAP services.

To display the definition and status of a specific ICAP service that is configured on the Content Engine, use the `show icap service service-name` EXEC command.

To display an ordered list of configured ICAP services and their status, use the `show icap vector-point vector-point-name` EXEC command.

```
ContentEngine# show icap vector-point ?
  reqmod-postcache  Display reqmod-postcache information
  reqmod-precache   Display reqmod-precache information
  respmod-precache  Display respmod-precache information
```

Displaying Statistics for ICAP Services

To display ICAP statistics for all of the configured ICAP services, use the **show statistics icap EXEC** command. This command has no arguments or keywords. There is no default behavior or values. The following is an example of the output of the **show statistics icap** command.

```
ContentEngine# show statistics icap
ICAP-client statistics (http proxy)
-----

Total requests for V1 via RPC:          0
Time per ICAP request (last 1k reqs):  0
ICAP daemon connection error:         0
Bad packets from ICAP daemon:         0
Error parsing HTTP req hdr from ICAP:  0
ICAP daemon internal error:           0

Total requests via outgoing proxy:     0
ICAP daemon overloaded:                0
Other errors:                          0
ICAP Daemon statistics
-----

Total requests served:                  0
Total requests served:                  0
Average latency in milliseconds:       0.000000
ICAP Service statistics
-----

Service -- servforicap
Service Errors:      0
Service Bypasses:   0
  Server -- icap://1.2.3.4/servforicap
    Total Reqmods (0), Total Respmods (0)
    Modifications (Reqmod - 0), (Respmod - 0)
    No Modifications (Reqmod - 0), (Respmod - 0)
    Error Responses (Reqmod - 0), (Respmod - 0)
    Server Errors:          0
    Server Bypasses:       0
    Options Req Success:   0
    Options Req Failed:    8569
    Max Conn Available    0
    Used Connections:      0
    Total Bytes sent:      0
    Total Bytes received:  0
    Total BPS sent:        0.000000
    Total BPS received:    0.000000
    Server State:          DISCONNECTED
ContentEngine#
```