Configuring ICAP on Standalone Content Engines

This chapter describes how to configure the Internet Content Adaptation Protocol (ICAP) on standalone Content Engines. The ACNS 5.2.1 software and later releases support ICAP for HTTP requests. The ACNS 5.4.1 software and later releases support ICAP for FTP-over-HTTP requests. Support for native FTP requests is not supported.

This chapter includes the following topics:

- Overview of ICAP, page 12-1
- Example of Configuring ICAP Services on a Content Engine, page 12-4
- Configuring ICAP for Standalone Content Engines, page 12-5
- Configuring Logging of ICAP Exchanges, page 12-9
- Displaying Information About an ICAP Configuration, page 12-10
- Displaying Statistics for ICAP Services, page 12-10

Note

For complete syntax and usage information for the CLI commands used in this chapter, see the Cisco ACNS Software Command Reference, Release 5.5 publication.

For information about how to configure ICAP for Content Engines that are registered with a Content Distribution Manager, see the Cisco ACNS Software Configuration Guide for Centrally Managed Deployments, Release 5.5.

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 as follows:

- 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.
Overview of ICAP

- 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.

**Note**
The ACNS 5.4.1 software and later releases also support ICAP for FTP-over-HTTP requests. Support for native FTP requests is not supported.

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

**About ICAP Performance**

With the response modification (respmod) vectoring point, which is used by virus-scanning ICAP vendors, the performance of the Content Engine model CE-7305 will be 300 transactions per second.

With the request modification (reqmod)-precache vectoring point, which is used by URL filtering ICAP vendors, the performance of the Content Engine model CE-7305 will drop 20 percent from the rated performance.

---

**Note**

The performance of the Content Engine will be limited by the performance of the ICAP server.

**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
Example of Configuring ICAP Services on a Content Engine

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

---

**Step 1** Specify which ICAP services should be performed on which requests that are received by the Content Engine by using the `icap apply {all | rules-template}` global configuration command.

For example:

- Instruct the Content Engine to run only the ICAP services that match the rules action `use-icap-service`.
  
  ```
  ContentEngine(config)# icap apply rules-template
  ```

- Instruct the Content Engine to run all of the ICAP services on all of the HTTP requests that it receives.
  
  ```
  ContentEngine(config)# icap apply all
  ```

**Step 2** Enable ICAP-related transaction logging, which is available in the local1/logs/icap/ directory.

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

**Step 3** 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
```

**Step 4** Define the ICAP service rules for this Content Engine by using the `rule` global configuration command.

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\.|.*datek\.|\.)"
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 enable 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 12-2.

Configuring ICAP for Standalone Content Engines

To configure ICAP on a standalone Content Engine, you must use the Content Engine CLI. The following sections describe the tasks you must complete:

- Configuring ICAP Settings for Standalone Content Engines, page 12-5
- Configuring ICAP Services on Standalone Content Engines, page 12-7
- Configuring an ICAP Server for Standalone Content Engines, page 12-8

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:

- 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, use the `icap append-x-headers` global configuration command.

- In the ACNS 5.1 software and later releases, 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 the ACNS 5.2 software and later releases, you can configure the Content Engine to append the username and group name 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 username and group name. Use the following two command options of the `icap append-x-headers` global configuration command, as follows:
  - Specify the `x-authenticated-user` option to allow 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.
- Specify the `x-authenticated-groups` option to allow 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.

To force strict rechecking of the cached content every time the ISTag changes, use the `icap bypass streaming-media` command. 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.

- In the ACNS 5.5.1 software and later releases, you can configure the Content Engine ICAP connection timeout.

To configure a timeout value for ICAP connections, use the `icap connection-timeout minutes` global configuration command.

Table 12-1 describes the `icap` global configuration command parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>append-x-headers</td>
<td>Appends x-headers during ICAP protocol handshake. Disabled by default. Can have multiple entries for various x-headers to be appended.</td>
</tr>
<tr>
<td>x-client-ip</td>
<td>Appends x-client-IP headers to the request that is sent to the ICAP server. Disabled by default.</td>
</tr>
<tr>
<td>x-server-ip</td>
<td>Appends x-server-IP headers to the request that is sent to the ICAP server. Disabled by default.</td>
</tr>
<tr>
<td>x-authenticated-user</td>
<td>Appends x-authenticated-user headers to the request that is sent to the ICAP server. Disabled by default.</td>
</tr>
<tr>
<td>x-authenticated-groups</td>
<td>Appends x-authenticated-groups headers to the request that is sent to the ICAP server. Disabled by default.</td>
</tr>
<tr>
<td>apply</td>
<td>Enables ICAP processing for HTTP and FTP-over-HTTP requests.</td>
</tr>
<tr>
<td>all</td>
<td>Enables ICAP processing for all HTTP and FTP-over-HTTP requests.</td>
</tr>
<tr>
<td>rules-template</td>
<td>Enables ICAP processing for HTTP and FTP-over-HTTP requests that match the Rules Template using the <code>rule action use-icap-service</code> global configuration command.</td>
</tr>
<tr>
<td>bypass</td>
<td>Enables bypassing of certain requested content.</td>
</tr>
<tr>
<td>streaming-media</td>
<td>Enables bypassing of streaming media.</td>
</tr>
<tr>
<td>connection-timeout</td>
<td>Sets the connection timeout for an ICAP transaction.</td>
</tr>
<tr>
<td>minutes</td>
<td>Number of minutes before the connection times out. The range is 1–480 minutes. The default is 20 minutes.</td>
</tr>
<tr>
<td>logging</td>
<td>Specifies logging-related options when ICAP services are used.</td>
</tr>
<tr>
<td>enable</td>
<td>Enables logging when ICAP services are used.</td>
</tr>
<tr>
<td>format</td>
<td>Specifies the logging format.</td>
</tr>
<tr>
<td>custom</td>
<td>Specifies a customized format for logging.</td>
</tr>
</tbody>
</table>
Configuring ICAP Services on Standalone Content Engines

In the ACNS 5.1 software and later releases, three vectoring points are supported to enable content adaptation, as described earlier in the “About ICAP Services and Vectoring Points” section on page 12-2. 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, regardless 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.

To configure ICAP services for a standalone Content Engine, use the \texttt{icap service} global configuration command:

\texttt{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 \| respmod_precache\} \| order order-number\}}

To configure a specific ICAP service, enter ICAP configuration mode by using the \texttt{icap service service-id} command.

Replace service-id with a name of your choice for the current ICAP service. When you enter the \texttt{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. \texttt{Table 12-2} describes the \texttt{icap service} command parameters for configuring an ICAP service on a standalone Content Engine.

\begin{table}[h]
\centering
\caption{Parameters for the icap service Command}
\begin{tabular}{|l|p{10cm}|}
\hline
Parameter & Description \\
\hline
service-id & Specifies a name of your choice for the current ICAP service. \\
\hline
enable & Enables ICAP services. \\
\hline
\end{tabular}
\end{table}

Table 12-2  Parameters for the icap service Command (continued)

<table>
<thead>
<tr>
<th>error-handling</th>
<th>Specifies error-handling options when an ICAP service is used.</th>
</tr>
</thead>
<tbody>
<tr>
<td>bypass</td>
<td>Bypasses this service when an error occurs with this service.</td>
</tr>
<tr>
<td>return-error</td>
<td>Returns an error message to the client and ends the request.</td>
</tr>
<tr>
<td>load-balancing</td>
<td>Specifies a load-balancing option for this service. See Table 12-3 for a list of these load-balancing options.</td>
</tr>
<tr>
<td>client-ip-hash</td>
<td>Allows for load-balancing among ICAP servers using the client IP address.</td>
</tr>
<tr>
<td>round-robin</td>
<td>Allows for round-robin load balancing among ICAP servers.</td>
</tr>
<tr>
<td>server-ip-hash</td>
<td>Allows for load balancing using the ICAP server IP address.</td>
</tr>
<tr>
<td>weighted-load</td>
<td>Allows for load balancing using a weight scheme that specifies weight on a server basis.</td>
</tr>
</tbody>
</table>

Table 12-3 describes the icap service load-balancing options.

Table 12-3  icap service Load-Balancing Options

<table>
<thead>
<tr>
<th>Load-Balancing Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client IP hash</td>
<td>Uses a hash-based algorithm based on the client IP address for load balancing the ICAP servers in the cluster.</td>
</tr>
<tr>
<td>Round-robin</td>
<td>Uses the round-robin method in which ICAP servers take turns processing HTTP and FTP-over-HTTP requests.</td>
</tr>
<tr>
<td>Server IP hash</td>
<td>Uses a hash-based algorithm based on the server IP address for load balancing among the ICAP servers in the cluster.</td>
</tr>
<tr>
<td>Weighted</td>
<td>Uses a farm of ICAP servers with different load capacities.</td>
</tr>
</tbody>
</table>

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 and FTP-over-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.

Table 12-4 describes the icap service server command parameters for configuring an ICAP server for a standalone 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 want to create transaction logs in ICAP’s standard logging format, choose `standard`.
- If you want to log additional fields not included in the standard format, choose `custom`.

**Note**
Customized format for transaction logging is not supported in the ACNS 5.1, ACNS 5.2.x, or ACNS 5.3.x software. Only standard transaction log format is available for ICAP services configured on a Content Engine.

### Table 12-4 Parameters for the `icap service server` Command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>server</td>
<td>Enables the ICAP server to be used for ICAP services using a URL.</td>
</tr>
<tr>
<td><code>url</code></td>
<td>URL based in the format <code>icap://ICAPserverIPaddress/servicename</code>.</td>
</tr>
<tr>
<td><code>keepalive-interval</code></td>
<td>(Optional). Specifies the keepalive interval. (This option was added in the ACNS 5.2 software.)</td>
</tr>
<tr>
<td><code>keepalive-interval</code></td>
<td>Keepalive interval in seconds. The default is 60 seconds. Valid values are from 1 to 3600 seconds.</td>
</tr>
<tr>
<td><code>max-connections</code></td>
<td>(Optional). Specifies the maximum number of connections to a particular ICAP server.</td>
</tr>
<tr>
<td><code>connection-number</code></td>
<td>Maximum number of connections. (The maximum is 5000.)</td>
</tr>
<tr>
<td><code>weight</code></td>
<td>(Optional) Sets weight percentage for load balancing if the weighted-load load-balancing scheme is used.</td>
</tr>
<tr>
<td><code>percentage</code></td>
<td>Percentage of the load that can be redirected to this ICAP server (0–100). This parameter must not be defined if you chose the <code>weighted-load</code> load-balancing scheme.</td>
</tr>
<tr>
<td><code>vector-point</code></td>
<td>Specifies vectoring point or content-processing mode to be used for the specified ICAP service.</td>
</tr>
<tr>
<td><code>reqmod-postcache</code></td>
<td>Allows modification of requests sent by the ICAP server and cached by the Content Engine.</td>
</tr>
<tr>
<td><code>reqmod-precache</code></td>
<td>Allows modification of requests as they are sent from the Content Engine to the ICAP server on their way to the origin server.</td>
</tr>
<tr>
<td><code>respmod-precache</code></td>
<td>Allows modification of requests after they return from the origin server.</td>
</tr>
<tr>
<td><code>order</code></td>
<td>Specifies the order for the content-processing mode used.</td>
</tr>
<tr>
<td><code>order-number</code></td>
<td>Order in which the content-processing mode handles requests to the Content Engine.</td>
</tr>
</tbody>
</table>
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-pprecache Display reqmod-pprecache information
  rspmod-pprecache Display rspmod-pprecache information
```

Displaying Statistics for ICAP Services

To display ICAP statistics for all of the configured ICAP services, enter the `show statistics icap` EXEC command. This command has no arguments or keywords. There is no default behavior or values.