

NetCache™ 5.4

Guide to ICAP Services

Network Appliance, Inc.
495 East Java Drive
Sunnyvale, CA 94089 USA
Telephone: +1 (408) 822-6000
Fax: +1 (408) 822-4501
Support telephone: +1 (888) 4-NETAPP
Documentation comments: doccomments@netapp.com
Information Web: <http://www.netapp.com>

Part number 210-00245
December 2002

Copyright and trademark information

Copyright information

Copyright © 1994–2002 Network Appliance, Inc. All rights reserved. Printed in the U.S.A.

Portions copyright © 1998–2001 The OpenSSL Project. All rights reserved.

No part of this book covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Portions of this product are derived from the Berkeley Net2 release and the 4.4-Lite-2 release, which are copyrighted and publicly distributed by The Regents of the University of California.

Copyright © 1980–1995 The Regents of the University of California. All rights reserved.

Portions of this product are derived from NetBSD, which is copyrighted by Carnegie Mellon University.

Copyright © 1994, 1995 Carnegie Mellon University. All rights reserved. Author Chris G. Demetriou.

Permission to use, copy, modify, and distribute this software and its documentation is hereby granted, provided that both the copyright notice and its permission notice appear in all copies of the software, derivative works or modified versions, and any portions thereof, and that both notices appear in supporting documentation.

CARNEGIE MELLON ALLOWS FREE USE OF THIS SOFTWARE IN ITS “AS IS” CONDITION. CARNEGIE MELLON DISCLAIMS ANY LIABILITY OF ANY KIND FOR ANY DAMAGES WHATSOEVER RESULTING FROM THE USE OF THIS SOFTWARE.

Software derived from copyrighted material of The Regents of the University of California and Carnegie Mellon University is subject to the following license and disclaimer:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notices, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment:

This product includes software developed by the University of California, Berkeley and its contributors.
4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS

INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Portions of the software were created by Netscape Communications Corp.

The contents of those portions are subject to the Netscape Public License Version 1.0 (the “License”); you may not use those portions except in compliance with the License. You may obtain a copy of the License at <http://www.mozilla.org/NPL/>.

Software distributed under the License is distributed on an “AS IS” basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is Mozilla Communicator client code, released March 31, 1998.

The Initial Developer of the Original Code is Netscape Communications Corp. Portions created by Netscape are Copyright © 1998 Netscape Communications Corp. All rights reserved.

Software derived from copyrighted material of Network Appliance, Inc. is subject to the following license and disclaimer:

Network Appliance reserves the right to change any products described herein at any time, and without notice. Network Appliance assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by Network Appliance. The use and purchase of this product do not convey a license under any patent rights, trademark rights, or any other intellectual property rights of Network Appliance.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

Trademark information

NetApp and the Network Appliance design are registered trademarks of Network Appliance, Inc. in the United States, Canada, and the European Union. Network Appliance is a registered trademark of Network Appliance, Inc. in Monaco and a trademark of Network Appliance, Inc. in the United States and Canada. FAServer is a registered trademark of Network Appliance, Inc. in the United States and the European Union. NetCache is a registered trademark of Network Appliance, Inc. in the European Union and Japan, and a trademark of Network Appliance, Inc. in the United States. SnapCopy is a registered trademark of Network Appliance, Inc. in the European Union and a trademark of Network Appliance, Inc. in the United States. SnapManager is a registered trademark of Network Appliance, Inc. in the United States. WAFL is a registered trademark of Network Appliance, Inc. in the United States, the European Union, and Canada. FilerView and SecureShare are registered trademarks of Network Appliance, Inc. in the United States. Data ONTAP is a trademark of Network Appliance, Inc. in the United States and Canada. Snapshot is a trademark of Network Appliance, Inc. in the United States and the European Union. NetApp—the Network Appliance Company is a registered trademark of Network Appliance, Inc. in the United States and other countries. ApplianceWatch, BareMetal, Center-to-Edge, ContentDirector, DataFabric, MultiStore, NearStore, SecureAdmin, Serving Data by Design, Smart SAN, SnapCache, SnapDrive, SnapMirror, SnapRestore, SnapVault, Vfiler, and Web Filer are trademarks of Network Appliance, Inc. in the United States.

Apple is a registered trademark and QuickTime is a trademark of Apple Computer, Inc. in the United States and/or other countries.

Microsoft is a registered trademark and Windows Media is a trademark of Microsoft Corporation in the United States and/or other countries.

RealAudio, RealNetworks, RealPlayer, RealSystem, RealText, and RealVideo are registered trademarks and RealMedia, RealProxy, and SureStream are trademarks of RealNetworks, Inc. in the United States and/or other countries.

Network Appliance is a licensee of the CompactFlash and CF Logo trademarks.

All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such.

Network Appliance NetCache is certified RealSystem compatible.

Table of Contents

	Preface	vii
Chapter 1	ICAP 1.0	1
	The ICAP 1.0 open standard protocol	2
	How ICAP 1.0 works	5
	ICAP 1.0 servicing for client requests	8
	Controlling access to ICAP services	11
	Configuring an ICAP 1.0 service farm	15
	ICAP logging and statistics	19
	Index	23

Preface

About this guide

This guide describes how to configure and optimize appliances running NetCache™ software. It provides information for the following supported NetCache platforms: C1100, C1105, C1200, C2100, C3100, and C6100.

For information about how to upgrade or service the system hardware, see the *Field Service Guide* for your NetCache appliance model.

Audience

This guide is for system administrators who are familiar with computer networks, Web technology, and HTTP services. This guide assumes that administrators have addressed ICAP server requirements prior to configuring NetCache support for ICAP services. This guide does not cover basic system administration topics such as IP addressing, routing, and network topology. It emphasizes the characteristics of the appliance hardware and the NetCache software.

Command and keyboard conventions

You can enter appliance commands on either the system console or from any client computer that can access the appliance through Telnet.

When describing key combinations, this guide uses the hyphen (-) to separate individual keys. For example, “Ctrl-D” means pressing the “Control” and “D” keys simultaneously.

This guide uses the term “type” to mean pressing one or more keys on the keyboard. It uses the term “enter” to mean pressing one or more keys and then Enter. Also, this guide uses the term “Enter” to refer to the key that generates a carriage return, although the key is named “Return” on some keyboards.

This guide uses capitalization and some abbreviations to refer to the keys on the keyboard. (The keys on your keyboard might not be labeled exactly as they are in this guide.)

Formatting conventions

The following table describes typographical conventions used in this guide.

Formatting convention	Type of information
<i>Italic type</i>	Words or characters that require special attention. Placeholders for information you must supply. For example, if the guide says to enter the <code>arp -d hostname</code> command, you enter the characters “arp -d” followed by the actual name of the host. Book titles in cross-references.
Monospaced font	Command and daemon names. Information displayed on the system console or other computer monitors. The contents of files.
Bold monospaced font	Words or characters you type. What you type is always shown in lowercase letters, unless it must be typed in uppercase letters to work properly.

About this guide

This guide discusses how ICAP 1.0 can extend network capability through increased support for value-added services. Additionally, it describes ICAP functionality, support for services, how to configure ICAP service farms, and how to implement policies for controlling client access.

Topics in this guide

This guide contains the following topics:

- ◆ [“The ICAP 1.0 open standard protocol”](#) on page 2
- ◆ [“How ICAP 1.0 works”](#) on page 5
- ◆ [“ICAP 1.0 servicing for client requests”](#) on page 8
- ◆ [“Controlling access to ICAP services”](#) on page 11
- ◆ [“Configuring an ICAP 1.0 service farm”](#) on page 15
- ◆ [“ICAP logging and statistics”](#) on page 19

The ICAP 1.0 open standard protocol

About ICAP 1.0

ICAP is an open standard protocol that extends the NetCache content delivery model by enabling value-added services. ICAP leverages the existing network infrastructure and makes it easy to implement third-party applications that provide services such as content filtering, virus scanning, language translation, and ad insertion.

Extending your network

ICAP functionality extends your network to provide content management services. It increases network efficiency and provides scalable solutions for processing Internet-based content. Because ICAP is an open protocol, you can extend ICAP servers to provide new services at the edge of any network. ICAP is extensible, efficient, and scalable.

Extensibility: The ICAP open standard allows you to choose applications without being locked into a proprietary, single solution for value-added services such as virus scanning.

Efficiency: Modified content is cached, eliminating the need to fetch a new copy or reprocess a request. The ICAP protocol directs or “vectors” specific content from NetCache to dedicated ICAP servers for adaptation. Vectoring frees up cache resources by pipelining process-intensive applications that consume bandwidth off the cache.

Scalability: ICAP is a scalable solution because deploying ICAP services is more efficient than adding a Web proxy to a network gateway for each service. In addition, several ICAP servers configured to provide the same type of service, such as, virus scanning, can service requests from a single cache.

Examples of services

NetCache and ICAP work together to provide the following types of services:

- ◆ Virus scanning—Check new content for virus contamination and provide cached content that was previously scanned.
- ◆ Content filtering—Filter and redirect an unauthorized or restricted request to another Web site.
- ◆ Ad insertion—Insert ads into Web pages or spawn new pages based on request preferences.

Virus scanning

Virus scanning services check Internet content for the presence of virus contamination. ICAP supports virus checking or scanning for HTTP and FTP content. When an ICAP server is installed and enabled in a caching system, all Web transactions are directed to the ICAP server. The ICAP server performs the virus checking service, and then returns the request to the NetCache appliance.

Note

ICAP virus scanning services are not supported for secure sockets (SSL). SSL is a secure connection, and data received over an SSL connection cannot be interpreted by NetCache or an ICAP server.

Content filtering

Content filtering services monitor Internet access to content by category, and redirect unauthorized or restricted requests to another Web site. Categorization provides a cache of URL-to-category mapping information that can be used in combination with NetCache ACLs to enforce category restrictions, that is, to control user requests for content. ICAP supports the following content filtering features:

- ◆ ICAP uses NetCache ACLs to enforce policy restrictions.
- ◆ ICAP uses request modification to dynamically query an ICAP content filtering service about Web site content. Results of the query can be cached, which eliminates the need to refetch the information.
- ◆ ICAP supports logging to track Web traffic through several parameters, such as user names, timestamps, and client IP addresses.

Ad insertion

Ad insertion services use click streams to profile users based on content retrieval. This information is then used for target advertising. Ad insertion techniques paste advertisements into Web pages as an effective means of connecting customers to products and merchants.

There are two ad insertion techniques: intrapage insertion and interpage insertion.

Intrapage ad insertion: With this technique, an advertisement is pasted directly into an HTML page. These types of banner ads direct users to various Web pages. This application requires content filtering so that only HTML pages are redirected to an ad insertion service.

Interpage ad insertion: With this technique, Web pages are inserted into the click/result stream at random intervals. The Web pages contain links to the original content. These types of ads are difficult or impossible to avoid by narrowing the browser window. They appear for 8 to 10 seconds, then users are redirected to the original link they wanted to access.

A targeted ad system can create ads tailored to the user. This type of ad system can identify a user by an IP address or authentication information to create a profile of the user. Access is recorded in ICAP log files so that ad vendors can obtain a count of how many times an ad has appeared.

How ICAP 1.0 works

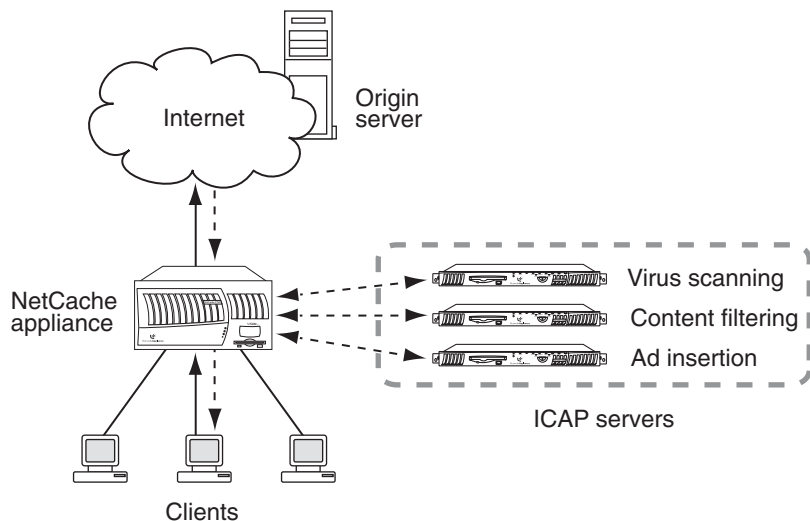
ICAP service farms

An ICAP service farm is a cluster of content adaptation servers through which NetCache routes requests. A NetCache appliance acts as an ICAP gateway to enable third-party functionality through ICAP service farms. All services in an ICAP service farm share the same functionality, for example, virus scanning. Each service farm, however, can support various types of the same service.

Vectoring for ICAP services

ICAP service farms are associated with a vector that designates the point at which an ICAP service is invoked. ICAP “vectors” client requests to service farms where requests and content are modified before being returned to the NetCache appliance.

The following illustration shows how NetCache vectors a client request to ICAP services.



Vectoring points

Request and response vectors are the points at which ICAP vectoring occurs within the NetCache appliance. Vectoring points are described as follows:

- ◆ Request vectors direct client requests to an ICAP server. The ICAP server modifies the request and returns it to the NetCache appliance. ICAP request

modification vectors include, REQMOD_PRECACHE and REQMOD_POSTCACHE.

- ◆ Response vectors direct client requests that have been processed by an origin server. NetCache redirects the response to the ICAP server for modification before content is delivered to the client. ICAP response modification consists of one vector, RESPMOD_PRECACHE.

Note

ICAP 1.0 forwards only HTTP and FTP content to precache response modification services (RESPMOD_PRECACHE).

REQMOD_PRECACHE (into the cache): Precache request modification services adapt a request on its way to the cache. An example is a filtering service that monitors Internet access to content by category and redirects unauthorized or restricted content. NetCache vectors the request to the ICAP server before it checks the cache for the requested object. The request is serviced and returned to the NetCache appliance. If the requested content exists in the cache, it has been adapted as a result of a previous request and is returned to the client. If the requested content does not exist in the cache, that is, if a cache miss occurs, the request is forwarded to the next ICAP server.

REQMOD_POSTCACHE (to the server): Postcache request modification services adapt a request on its way to an origin server. Content is serviced for cache misses only.

RESPMOD_PRECACHE (from the server): Precache response modification services adapt a response that is sent from an origin server. The ICAP server listens for ICAP requests from Web servers. The requests are scanned, and content is adapted before it is cached. Examples of these services are, virus scanning, ad insertion, and language translation.

See “[ICAP 1.0 servicing for client requests](#)” on page 8 for an illustrated example that maps client requests through the vectoring points of an ICAP-enabled NetCache appliance.

Ordering of services

Each ICAP service is associated with the vectoring point that is specified when the service farm is created. For example, an ordered list of services for the RESPMOD_PRECACHE vector might look like the following:

1. Virus checking
2. Language translation

3. Ad insertion

The order of invocation for services in an ICAP pipeline, for example, a pipeline that consists of both language translation and virus scanning, is determined by the relative order of the service within the list.

Load balancing

A load-balancing scheme for ICAP services is required; therefore, one load-balancing scheme must always be in effect. Load-balancing schemes determine which server a request is sent to, and different schemes apply to different scenarios.

Load-balancing schemes include the following:

- ◆ Round robin
- ◆ IP address
- ◆ Least usage (LU)

Round robin: In this scheme, the servers are accessed in rotation. Each new request is sent to the next available server, in sequence. Round robin is especially useful for filtering, for homogeneous groups of machines, and for processes that are not compute-intensive. The default load-balancing scheme is round robin.

IP address: The IP address scheme load balances using IP address hashing to ensure that a request is sent to the same ICAP service each time. This scheme can be useful for bandwidth monitoring or for advertising in order to monitor a trail through accessed sites.

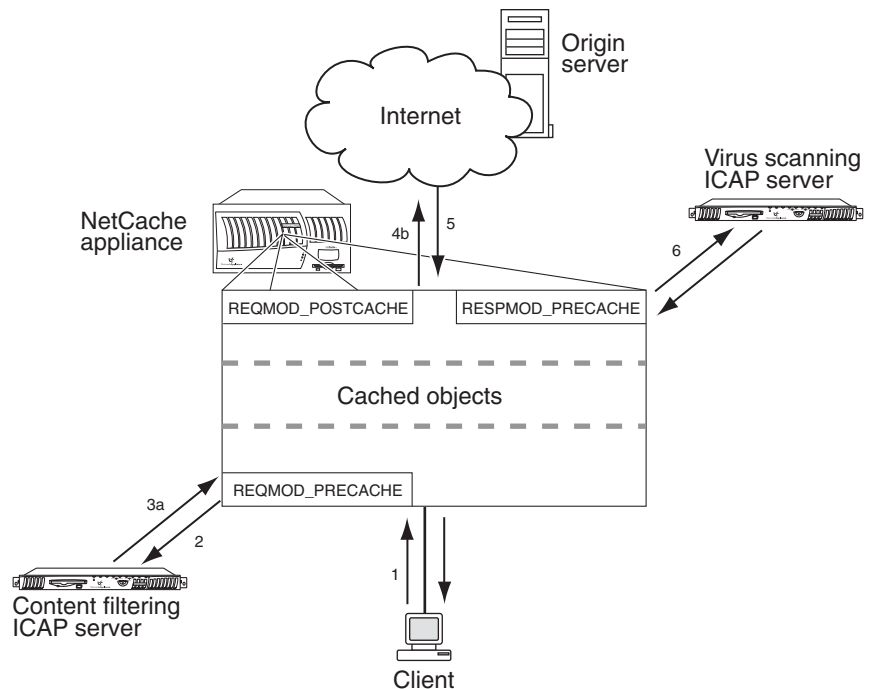
Least usage: The least usage scheme load balances based on a combination of service capability and the number of outstanding connections. Service capability is measured using an average data process rate, which is derived using the number of outstanding transactions for each host. The higher the average data process rate or the lower the number of outstanding transactions, the more likely it is that a service will be selected. This scheme is useful for heterogeneous groups of machines, virus checking, and compute-intensive processes with long processing times.

Support for NetCache ACLs

NetCache ACLs are used to implement policies and control access to services. ACLs must be enabled as part of configuring ICAP services, and rules must be applied to direct ICAP requests. For more information, see [“Controlling access to ICAP services”](#) on page 11.

ICAP 1.0 servicing for client requests

Client request flow The following illustration maps a client request for services through an ICAP-enabled NetCache appliance that is configured to provide content filtering and virus-scanning services.



The following table describes the flow of the client request shown in the preceding illustration.

Stage	Description
1	The client makes a request for content.
2	NetCache vectors the request to REQMOD_PRECACHE, and the request is passed to the ICAP content-filtering server.

Stage	Description
3	The request is returned to the NetCache appliance as one of the following: a. A potentially modified request or URL redirect b. An error response that is returned to the client
4	NetCache checks its cached objects for the requested content. a. If the request for content is a cache hit (that is, if the content exists in the cache), the cached object is the result of a previous RESPMOD_PRECACHE service, such as virus scanning, and can be returned safely to the client. b. If the request for content is a cache miss (that is, if the content does not exist in the cache), the request is then forwarded to the origin server.
5	The origin server satisfies the request for content and returns it to the NetCache appliance.
6	NetCache vectors the request to RESPMOD_PRECACHE, and the request is passed to the ICAP virus-scanning service farm. The content is checked for contamination before being returned to the NetCache appliance, where the object is cached and returned to the client.

Cache indexing

NetCache support for multiple ICAP services introduces the possibility that different services will process the same request, and that similar requests will be processed differently. Cache indexing is the mechanism by which NetCache ensures that the content requested is the content that is served. Services such as language translation can process similar requests and allow different users to receive unique content. For example, if two users request the same content or URL, and the content is translated to French and German, both translated versions of the same URL are cached by the same NetCache appliance.

NetCache uses cache indexing to identify all RESPMOD_PRECACHE transformations in order to serve the appropriately modified content. Typically, cached objects are indexed on the basis of origin server URL. However, to support ICAP services, NetCache now stores origin server URLs and ICAP

URLs. This information is a combination of the object and the services performed for that request. Cache indexing ensures that if the URL and the RESPMOD services are the same, NetCache serves a cache hit.

Controlling access to ICAP services

ACLs

NetCache access control lists (ACLs) are sets of rules that create policies for controlling access to services. ACL entries (or rules) allow, deny, or otherwise act on requests based on variables such as client IP addresses. NetCache is permissive by default, that is, ACLs are not enabled and no requests are denied.

ICAP services are configured using ACLs. ACLs must be enabled to perform ICAP services, and rules must be applied to direct user requests. You can use the Appliance Manager to enable ACLs for ICAP and to create rules that

- ◆ Direct requests for ICAP services
- ◆ Control who receives adapted content

How ACLs work

Actions are combined with expressions to form rules. Variables and associated values can be used to form pattern-matching expressions. When processing a request, NetCache compares the request against pattern-matching expressions until a match occurs. When a match occurs, actions are executed. If no match occurs, the action is not executed and rule checking continues.

If you are not familiar with ACLs in NetCache and writing rules to control client access, see the *Security Guide* to form a comprehensive understanding of how to create rules and implement policies for controlling access to the NetCache appliance.

Creating rules

Rules are an ordering of variables in the form of an expression. Expressions specify actions for directing ICAP requests. Rules for controlling ICAP services list a set of ICAP service farms. The ACL syntax is insensitive to the order in which services are listed. For example, the following rules are equivalent:

```
icap(virus_check, ad_insert) user bob
```

```
icap(ad_insert, virus_check) user bob
```

The actual order of services is determined during ACL processing, when the vectoring point for each service retrieves its order from the services list. For more information, see “[Ordering of services](#)” on page 6.

ACL actions for controlling ICAP services

NetCache supports the following actions and variables that are specific to handling requests for ICAP services.

Action/Variable	Description
icap	Directs requests for ICAP services. For example: <pre>icap(virus_check) server-ip 10.20.30.151 and url matches exe</pre> In this example, the virus-scanning service is configured for all HTTP requests to server IP 10.20.30.151 that contain a URL pattern that matches “exe.” Note_____ When listing multiple services, you must separate each with a comma followed by a blank character. For example, <code>icap(service1, service2, service3)</code>. _____
icap_now	Sends a precache request modification (REQMOD) to an ICAP service. The <code>icap_now</code> action is used for ICAP services that must be invoked early in an ACL evaluation process, for example, content-filtering services. Filtering services typically implement access control policy checks that must be invoked during ACL processing. For example: <pre>icap_now(content_filter)</pre> In this example, ICAP filtering is invoked for all requests. Note_____ All <code>icap_now</code> service farms must provide precache request modification services (REQMOD_PRECACHE vector), which adapt a request that is sent to the cache, for example, content filtering. See “Vectoring points” on page 5 for a description of request and response modification service types. _____

Action/Variable	Description
attribute	Applies to attributes returned by ICAP services. The <code>attribute</code> variable directs NetCache to compare values, such as a URL category returned by an ICAP service during an ACL evaluation. For example: <pre>attribute service_farm_name attribute_value</pre> In this example, the named service returns the attribute, and the value is compared. Note_____ An ICAP service must be invoked before a check for its attribute can be made. The <code>icap_now</code> action advances ACL evaluation for services. If an <code>icap_now(service_farm_name)</code> clause is not processed before an attribute clause, the service is not recognized, and an error message is returned to the client. _____

Content category restriction using NetCache ACLs

The NetCache access control scheme provides a mechanism for setting content category-based restrictions. NetCache ACL settings can be used to allow or deny user access to content.

Example: The `icap_now` action enables NetCache to send an ICAP precache request modification early in an ACL evaluation so that subsequent ACLs can be applied to categories. Filtering services typically categorize content, and attribute values can be used to specify access to content. Consider the following rule:

```
icap_now(content_categorization) any
```

```
deny user bob and attribute content_categorization sports
```

In this example, the `icap_now` action enables NetCache to send an ICAP precache request modification early in an ACL evaluation. the `attribute` variable is used for category-based control. The `icap_now` action invokes the ICAP service, and user Bob is denied access to all Web sites for which the ICAP service returns the attribute `sports`.

Authenticating users

Authentication is the process of determining a users identity. NetCache is permissive by default, which means that authentication is not enabled and access to all protocols is allowed. The NetCache authentication and ACL features enable you to control access to the NetCache appliance by individual user or groups of

users. Authentication verifies the identity of the user, and ACLs authorize access to services. Use the options on the Authentication pages of the Appliance Manager to enable authentication (Setup tab > Authentication).

If you are not familiar with authentication in NetCache, see the *Security Guide* to form a comprehensive understanding of supported methods for authenticating users to NetCache services.

Using ACLs to authenticate users

NetCache ACL settings can be used to force authentication before user requests are serviced. The following actions can be used to form rules that control requests for access to supported services.

Action	Description
auth	Require authentication and continue the rule checking sequence if already authenticated.
re-auth	Force reauthentication.
try-auth	Require authentication and continue the rule checking sequence even if authentication fails.

Example: ACLs can be used to create a policy that authenticates users and directs access to ICAP-serviced content. Consider the following policy:

```
auth any
icap(translate-french) group engineering
icap(content-filter) not group marketing
deny attribute content-filter news and time 9am-5pm
```

In this example, NetCache directs requests from the engineering group to the ICAP language translation server. With the exception of requests from the marketing group, all requests are sent to the ICAP content-filtering server. The ICAP filtering service categorizes content, and all filtered requests are denied access to news Web sites from 9:00 a.m. to 5:00 p.m.

Configuring an ICAP 1.0 service farm

You must plan the configuration

ICAP service farm configuration will require less time if you make some decisions and gather some information before you start configuring the NetCache appliance.

To plan for the ICAP service farm configuration, you must do the following:

- ◆ Obtain capacity planning information from the service vendor to ensure that your compute resources meet service requirements.
- ◆ Configure the NetCache ICAP client and the ICAP server on the same network to ensure best performance.
- ◆ Ensure that your ICAP server is appropriately configured with a third-party service, such as virus scanning.
- ◆ Determine the appropriate vectoring point for the ICAP services you intend to support. For example, content filtering is REQMOD_PRECACHE. See “[Vectoring points](#)” on page 5 for a description of vectors and their relationship to ICAP services.
- ◆ Determine a load-balancing scheme: round robin, IP address, or least usage. See “[Load balancing](#)” on page 7 for a description of load-balancing schemes for ICAP service farms.
- ◆ Determine your ICAP service URL. Service farms can support various services, and each service is associated with a URL. The URL string identifies the service to NetCache and toggles the ICAP service on or off. For example:

```
icap:// IP address or hostname of the ICAP server : port /  
service name [on | off]
```

Note

The NetCache default port number for ICAP services is 1344.

- ◆ Understand how to enable and apply ACLs for ICAP services. ACLs must be enabled, and rules to direct requests and control services are required. For example:

```
icap(virus_check, ad_insert) user bob
```

See “[Controlling access to ICAP services](#)” on page 11 for information about handling requests for ICAP services.

Configuring ICAP using the CLI

To get information about ICAP 1.0 configuration options from the CLI, enter the following command:

```
set config.icapv1.?
```

The preceding command lists the available ICAP options.

To get help for a specific ICAP option, enter the following command:

```
help config.icapv1.optionname
```

Creating an ICAP service farm

Use the Appliance Manager to add or delete ICAP service farms. When service farms are added or removed, the service farm data structure is updated and reflected in the Appliance Manager. No reboot is necessary after an ICAP service farm is added or edited. When a host is deleted from the service farm list, the ICAP service is removed, all connections are shut down, and eventually cached items become stale and are reclaimed.

To create an ICAP service farm, complete the following steps.

Step	Action
1	In the Appliance Manager, select Setup tab > ICAP > ICAP 1.0.
2	Click the Service Farms tab, then click New Service Farm.
3	Enter the ICAP service farm name.
4	Select the vectoring point within the ICAP pipeline where ICAP services are invoked: ◆ REQMOD_PRECACHE modifies a request before it is sent to the cache. ◆ REQMOD_POSTCACHE modifies a request before it is sent to the origin server. ◆ RESPMOD_PRECACHE modifies objects sent from an origin server on the way to the cache.
5	Specify the order in which the service is listed. New farms are added to the end of the list of ICAP services.
6	Select the Service Farm Enable check box to enable the service farm.

Step	Action
7	In the Load Balancing option, select a load-balancing scheme: ◆ Round robin uses a one-to-one rotation among service farms. ◆ IP-address-based uses IP address hashing to ensure that the same clients fetch through the same ICAP service. ◆ Least usage selects a host based on a combination of service capability and the number of outstanding connections.
8	Select the Bypass on Failure check box if you want to bypass the service farm when an ICAP server is unavailable.
9	Specify the consistency or service availability within an ICAP service farm if you are running more than one version of a service. ◆ <i>Weak</i> specifies a service farm that runs more than one version of a service. Weak is the preferred consistency setting because content is not checked for inconsistent results. ◆ <i>Strong</i> specifies a service farm that should not be configured to run more than one version of a service. Strong deactivates service versions that are not consistent with the majority. ◆ <i>lbw</i> (load-balanced weak) specifies a service farm configuration that is compromised if the number of servers is reduced. lbw toggles to Weak if servers become overloaded.
10	Specify the lbw threshold as a value between 0 and 1. This value is required only if lbw is specified as a consistency setting. This is the number of ICAP services that can operate at greater than peak server load.

Step	Action
11	In the Services option, specify the ICAP service URLs. Use the following format: <pre>icap:// IP address or hostname of the ICAP server : port / service_name [on off]</pre> where <i>IP address or hostname</i> specifies the service, <i>port</i> specifies the default ICAP port, and <i>service_name</i> (optional) specifies the name of a service. Enable or disable the service by specifying “on” or “off” at the end of the URL string. Note_____ The NetCache default port number for ICAP services is 1344. _____
12	Click Commit Changes at the bottom of the page.

Editing a service farm

To edit an existing service farm, complete the following steps.

Step	Action
1	In the Appliance Manager, select Setup tab > ICAP > ICAP 1.0.
2	Click the Service Farms tab.
3	Select or clear the Enable check box for a specific service farm to enable or disable it.
4	Select or clear the Del check box for a specific service farm to delete or retain it.
5	To edit a service farm, click Edit for the service farm you want to edit.
6	Change any options on the Edit ICAP Service Farm page, as required. For more information about the various options, see “Creating an ICAP service farm” on page 16.
7	Click Commit Changes at the bottom of the page.

ICAP logging and statistics

About logging

The ICAP log is a record of ICAP server transactions. You can specify one of two ICAP log file formats: default or custom. The default ICAP log file format captures common errors, such as unavailable services or ACL configuration problems. You can configure custom ICAP log files to record selected services, object size, and result codes for each ICAP transaction.

ICAP default log file format

Entries in the ICAP log file reflect the timestamp, time taken, client IP address, bytes available, requested URL, pipeline of ICAP service farm names, pipe times, and authenticated user name for a request. For example:

```
x-timestamp time-taken c-ip bytes cs-url x-icap-pipenames x-icap-pipetimes x-username
```

The following is an example of the default ICAP log file format:

```
984170515.26 0.332 173.20.100.116 23134  
http://www.cnn.com/sports|(virusblocker)|adinsert [0.100]|0|.232  
paul
```

A pipe (|) demarcates service farm names. Parentheses that enclose server farm names designate a bypass for requests in the event that services are unavailable, and angle brackets (< >) designate no bypass in the event that services are unavailable. Square brackets ([]) indicate that the vectored request, REQMOD or RESPMOD, resulted in a cache hit. Pipeline entries are followed by a record of the time taken by a service farm to perform a service.

ICAP custom log file settings

You can apply default attributes and customize log file entries for each ICAP transaction. Custom settings include the x-icap-categories and x-icap-pipeservs attributes:

- ◆ The x-icap-categories attribute can be used to display content filter categories, such as sports and news. For example, a custom log that records content categories is useful for identifying user-requested content.
- ◆ The x-icap-pipeservs attribute can be used to display pipeline service URLs. For example, a custom log that records pipelines can be useful for troubleshooting a slow service.

Enabling ICAP logging

To enable ICAP logging, complete the following steps.

Step	Action
1	In the Appliance Manager, select Setup tab > ICAP > ICAP 1.0.
2	Select the General tab.
3	Select the ICAP 1.0 Log Enable check box.
4	Select the log format: default or custom. If you select custom, edit the text box to include the log attributes that you want recorded.
5	Click Commit Changes at the bottom of the page.

Note

Log file switching and configuration options are located on the System - Logging page of the Appliance Manager (Setup tab > System > Logging).

Viewing the log file

To view the log file, complete the following steps.

Step	Action
1	In the Appliance Manager, select Data tab > Log Files > ICAP 1.0.
2	Select a log file from the list of log files.
3	Select the log file action. You can do one of the following: ◆ View the file. ◆ View the last <i>n</i> lines of the file. Enter the number of lines in the text box. ◆ Delete the log file. ◆ Copy the file to your computer as is or in gzip compressed format. ◆ Send the file to an FTP or Web server. Specify the server using the following format: [ftp: http:] + //user:password@server:port/path/file
4	Click Commit Changes at the bottom of the page.

Statistics

ICAP statistics are reported on a per-farm, per-server basis. These statistics enable you to monitor the status of active ICAP service farms. ICAP statistics are a collective report of service and processing values, such as the following:

- ◆ Active service connections to the NetCache ICAP connection cache
- ◆ Bytes sent between NetCache and ICAP services
- ◆ Total number of services in a farm
- ◆ Approximate number of requests sent to a particular ICAP service
- ◆ Total number of failed attempts to initialize or open a connection to an ICAP service

ICAP statistics can be viewed on the ICAP - Status page of the Appliance Manager (Data tab > ICAP > Status).

Index

A

ACLs

- actions and variables 12
- authenticating users 14
- controlling access to services 12
- creating rules 11
- description of 11
- ordering services 11
- restricting content 13

ad insertion 3

authentication, user 13

C

cache indexing 9

configuring

- ICAP log files 20
- ICAP service farms 16

content filtering 3

I

ICAP 1.0

- ACLs 7
- authenticating users 13
- configuring 15
- description of 2
- load balancing 7
- logging 19
- ordering services 6
- planning to configure 15
- statistics 21
- supported services 2

L

load balancing

- IP address 7
- least usage 7

round robin 7

logging

- custom entries 19
- default log file format 19
- description of 19
- enabling 20
- viewing log files 20

O

ordering ICAP services 6

P

port, ICAP default 15

S

service farms

- creating 16
- description of 5
- editing 18

services

- ad insertion 3
- content filtering 3
- service order 6
- virus scanning 3

statistics 21

V

vectoring points

- REQMOD_POSTCACHE 6
- REQMOD_PRECACHE 6
- request vector 5
- RESPMOD_PRECACHE 6
- response vector 6

virus scanning 3

