Symantec™ Scan Engine Software Developer's Guide

Symantec™ Scan Engine Software Developer's Guide

Documentation version 5.2

Legal Notice

Symantec Corporation
20330 Stevens Creek Blvd.
Cupertino, CA 95014

http://www.symantec.com

Technical Support

Symantec Technical Support maintains support centers globally. Technical Support's primary role is to respond to specific queries about product feature and function. The Technical Support group also authors content for our online Knowledge Base. The Technical Support group works collaboratively with the other functional areas within Symantec to answer your questions in a timely fashion. For example, the Technical Support group works with Product Engineering and Symantec Security Response to provide alerting services and virus definition updates.

Symantec's maintenance offerings include the following:

- A range of support options that give you the flexibility to select the right amount of service for any size organization
- A telephone and web-based support that provides rapid response and up-to-the-minute information
- Upgrade assurance that delivers automatic software upgrade protection
- Global support that is available 24 hours a day, 7 days a week
- Advanced features, including Account Management Services

For information about Symantec's Maintenance Programs, you can visit our Web site at the following URL:

www.symantec.com/techsupp/

Contacting Technical Support

Customers with a current maintenance agreement may access Technical Support information at the following URL:

www.symantec.com/techsupp/

Before contacting Technical Support, make sure you have satisfied the system requirements that are listed in your product documentation. Also, you should be at the computer on which the problem occurred, in case it is necessary to recreate the problem.

When you contact Technical Support, please have the following information available:

- Product release level
- Hardware information
- Available memory, disk space, and NIC information
- Operating system

- Version and patch level
- Network topology
- Router, gateway, and IP address information
- Problem description:
  - Error messages and log files
  - Troubleshooting that was performed before contacting Symantec
  - Recent software configuration changes and network changes

Licensing and registration

If your Symantec product requires registration or a license key, access our technical support Web page at the following URL:

www.symantec.com/techsupp/

Customer service

Customer service information is available at the following URL:

www.symantec.com/techsupp/

Customer Service is available to assist with the following types of issues:

- Questions regarding product licensing or serialization
- Product registration updates such as address or name changes
- General product information (features, language availability, local dealers)
- Latest information about product updates and upgrades
- Information about upgrade assurance and maintenance contracts
- Information about the Symantec Buying Programs
- Advice about Symantec's technical support options
- Nontechnical presales questions
- Issues that are related to CD-ROMs or manuals

Maintenance agreement resources

If you want to contact Symantec regarding an existing maintenance agreement, please contact the maintenance agreement administration team for your region as follows:

- Asia-Pacific and Japan: contractsadmin@symantec.com
- Europe, Middle-East, and Africa: semea@symantec.com
- North America and Latin America: supportsolutions@symantec.com

Additional enterprise services

Symantec offers a comprehensive set of services that allow you to maximize your investment in Symantec products and to develop your knowledge, expertise, and global insight, which enable you to manage your business risks proactively. Enterprise services that are available include the following:

Symantec Early Warning Solutions	These solutions provide early warning of cyber attacks, comprehensive threat analysis, and countermeasures to prevent attacks before they occur.
Managed Security Services	These services remove the burden of managing and monitoring security devices and events, ensuring rapid response to real threats.
Consulting Services	Symantec Consulting Services provide on-site technical expertise from Symantec and its trusted partners. Symantec Consulting Services offer a variety of prepackaged and customizable options that include assessment, design, implementation, monitoring and management capabilities, each focused on establishing and maintaining the integrity and availability of your IT resources.
Educational Services	Educational Services provide a full array of technical training, security education, security certification, and awareness communication programs.

To access more information about Enterprise services, please visit our Web site at the following URL:

www.symantec.com

Select your country or language from the site index.

Contents

Chapter 1

Getting started

This chapter includes the following topics:

About Symantec Scan Engine

Symantec Scan Engine, formerly marketed as Symantec AntiVirus™ Scan Engine, is a carrier-class, network-accessible, content-scanning engine. Symantec Scan Engine provides content scanning capabilities to any application on an IP network, regardless of platform.

Symantec Scan Engine features all of the key content-scanning technologies that are available in the complete line of Symantec products. Symantec Scan Engine is one of the most effective solutions available for protecting your network against a variety of undesirable content.

For more information, see the *Symantec Scan Engine Implementation Guide*.

About the software developer's guide

Software developers can use the information that is provided in the *Symantec Scan Engine Software Developer's Guide* to create client applications that let

third-party applications integrate with Symantec Scan Engine for a variety of content scanning services. You can configure client applications to communicate with Symantec Scan Engine using one of several supported protocols. However, the only protocol that is supported in the software developer's guide is the Internet Content Adaptation Protocol (ICAP) version 1.0, as presented in RFC 3507 (April 2003).

What's new

If you have constructed your own ICAP 1.0 client for Symantec Scan Engine, you can take advantage of the following new features:

Ability to construct an ICAP client connector using Java API	If your environment has Java, you can use the Java API plug-in (SymJavaAPI.jar) to integrate with Symantec Scan Engine. The Java API provides client antivirus scanning and repair services using the ICAP protocol. The Java API supports the FILEMOD and RESPMOD scanning modes, and it contains the built-in ability to stream files. For more information, see Scan_Engine_SDK/Java/Docs/SymJavaAPIDocs.jar on the product CD.
Ability to construct an ICAP client connector using .NET API	If your environment has .NET Framework, you can use the .NET API plug-in (symcsmsnetapi.dll) to integrate with Scan Engine. The .NET API provides client antivirus scanning and repair services using the ICAP protocol. The .NET API supports the FILEMOD and RESPMOD scanning modes, and it contains the built-in ability to stream files. For more information, see Scan_Engine_SDK/C#/Docs/SymCSharpAPIDocs.chm on the product CD.
Support for gcc 3.4.6	Symantec Scan Engine software developer's kit now provides support for gcc-3.4.6 for SPARC Solaris 10 and Red Hat Enterprise Linux 3.

See "Scanning non-HTTP data (RESPMOD and FILEMOD)" on page 63.

About integrating with Symantec Scan Engine

Software developers can create client applications (connectors) that let third-party applications integrate with Symantec Scan Engine for content scanning services using ICAP as the communication protocol.

You can create a custom integration using any of the following methods:

Use the basic client-side antivirus application program interface (API) C library.	If you plan to integrate antivirus scanning only, you can use the basic antivirus API. URL filtering is not available using the antivirus API. The antivirus API includes 12 libraries which includes static and dynamic libraries for each supported platform. The API library consists of 10 functions that provide scanning and repair services to client applications.
Construct your own ICAP 1.0 client for the Symantec Scan Engine.	If you construct your own ICAP client, you can specify whether to perform antivirus scanning and URL filtering for outgoing and incoming requests.

About ICAP

ICAP is a lightweight protocol that was originally created to execute a remote procedure call on HTTP messages. ICAP is part of an evolving architecture that lets corporations, data communication companies, and Internet service providers (ISPs) dynamically scan, change, and augment data as it flows through ICAP servers. The protocol lets ICAP clients pass data to ICAP servers for adaptation (some type of transformation or other processing, such as virus or URL filtering). The server executes its transformation service on the data and responds to the client, possibly with modified content.

In a typical integration for processing HTTP traffic, a caching proxy server retrieves the requested information from the Web. At the same time, it caches the information (stores a copy on disk) and, where possible, serves multiple requests for the same Web content from the cache. A caching proxy server can use ICAP to communicate with Symantec Scan Engine and request that content that is retrieved from the Web be scanned and repaired, if necessary.

About the C API

You can use the client-side API C library to configure an application to pass files to Symantec Scan Engine for scanning. The API includes 16 libraries, which includes static and dynamic libraries for each supported platform. The API library

consists of 12 functions that provide scanning and repair services to client applications.

The Symantec Scan Engine API C library is available for the following platforms:

- Red Hat Linux 7.2 and later (using either gcc 2.95.2 or 3.2)
- Red Hat Enterprise Linux 3 (using gcc 3.4.6)
- Solaris 7 and later (using either gcc 2.95.2 or 3.2)
- Solaris 10 (using gcc 3.4.6)
- Windows 2000 Server/Advanced Server (using either Microsoft Visual Studio 6 or 7)

A C header file is included.

About licensing

Key features for Symantec Scan Engine are activated by license. After you install Symantec Scan Engine, you install licenses through the Symantec Scan Engine administrative interface.

The content scanning features, including antivirus and URL filtering, are activated by product licenses. Subscription licenses let you obtain updates to virus definitions and URL filtering content updates. When a license expires, a new license must be installed.

When no product license is installed, Symantec Scan Engine is not operational. After you install a product license, you can access the relevant portions of the administrative interface, and Symantec Scan Engine is operational. For example, if you activate a product license for antivirus scanning only (with no URL filtering), you are not able to access those portions of the administrative interface that relate to URL filtering.

When no subscription license is installed or a subscription license expires, Symantec Scan Engine is operational, but updates are not permitted. New virus definitions updates are not downloaded to keep protection current, and URL-filtering updates to the URL lists and DDR dictionaries are not permitted.

For more information about licensing, see the *Symantec Scan Engine Implementation Guide*.

Considerations for implementation

Symantec Scan Engine can be easily implemented in an existing infrastructure. Symantec Scan Engine runs on Sun Solaris, Red Hat Linux, SuSE Linux, and

Windows Server platforms. You can run Symantec Scan Engine on the same computer or a different computer than the client application.

For more information about Symantec Scan Engine system requirements, see the *Symantec Scan Engine Implementation Guide*.

How to implement scanning

You can create a custom integration in any of the following ways using ICAP:

Use the basic client-side antivirus application program interface (API) C library.	If you plan to integrate antivirus scanning only, you can use the basic antivirus API. URL filtering is not available using the antivirus API. The antivirus API includes 12 libraries, which includes static and dynamic libraries for each supported platform. The API library consists of 10 functions that provide scanning and repair services to client applications. The Symantec Scan Engine API supports both file-based and stream-based scanning. See "File-based scanning" on page 73. See "Stream-based scanning" on page 73.
Construct your own ICAP 1.0 client for Symantec Scan Engine.	If you construct your own ICAP client, you can specify whether to perform antivirus scanning and URL filtering for outgoing and incoming requests.

How to maximizing performance

In a typical configuration, files are passed to Symantec Scan Engine through a socket over the network because Symantec Scan Engine is running on a separate computer. Depending on the network setup, client applications (applications that are configured to pass files to Symantec Scan Engine for scanning) can pass a full path rather than the actual file to Symantec Scan Engine for improved performance. For example, files to be scanned might be located on a drive that can be mounted over the network, such as a shared drive in Windows or a network file system (NFS) drive. If the client application and Symantec Scan Engine have access to a shared directory, the client application can place the file in the shared directory and pass the full path to Symantec Scan Engine.

For cases in which the client application is running on the same computer as Symantec Scan Engine, the client application can pass the file name to Symantec

Scan Engine, and Symantec Scan Engine can open the file and scan it in place on the computer.

See "Local file scanning (FILEMOD)" on page 64.

About Symantec Scan Engine API load balancing

The Symantec Scan Engine API provides scheduling across any number of computers that are running Symantec Scan Engine. Client applications that pass files to Symantec Scan Engine benefit from load-balanced virus scanning without any additional effort. When multiple scan engines are used, the API determines the appropriate scan engine to receive the next file to be scanned. The determination is based on the scheduling algorithm.

Another scan engine is called when any of the following events occur:

- A scan engine is unreachable
- A scan engine stops responding during a scan
- If you use ICAP and the queued requests threshold is reached
  When the threshold is reached, Symantec Scan Engine rejects the scan request.

The faulty scan engine is taken out of rotation for a period of time (30 seconds is the default). If all of Symantec Scan Engines are out of rotation, the faulty scan engines are called again. The API does not stop trying to contact Symantec Scan Engines unless five engines do not respond. Or it appears that a file that is being scanned might have caused more than one engine to stop responding.

Where to start

Configuring client applications to use ICAP 1.0 to pass files to Symantec Scan Engine for scanning involves the following process:

- Become familiar with the design and features of the software.
  See also the *Symantec Scan Engine Implementation Guide*.
- Decide how to deploy Symantec Scan Engine to meet your specific requirements.
  See "Considerations for custom integration" on page 19.
- Install and configure Symantec Scan Engine to use ICAP as the communication protocol.
  For more information, see the *Symantec Scan Engine Implementation Guide*.
- Install the API libraries.
  See "About the C API" on page 13.

- Configure the client applications that will send files to Symantec Scan Engine for scanning.

Chapter 2

Configuring Symantec Scan Engine for custom integrations

This chapter includes the following topics:

Considerations for custom integration

Symantec Scan Engine is designed to be integrated into any environment to provide content scanning for any application. Client applications are configured to pass files to Symantec Scan Engine. Symantec Scan Engines performs the required functions and returns the request with the necessary changes. Symantec Scan Engine supports custom integrations in which the software developer creates a client application (connector) to provide content scanning services for a third-party application. The client application communicates with Symantec Scan Engine using ICAP 1.0.

You must configure Symantec Scan Engine to support the custom integration as follows:

- Selecting ICAP as the communication protocol
- Configuring ICAP-specific options
- Configuring the scanning parameters

You must decide how to configure the client application and Symantec Scan Engine to ensure that scanning is handled appropriately. This decision can depend on the capabilities of the third-party application.

For example, for antivirus scanning, the client application can decide which file types to scan and pass only the appropriate files to Symantec Scan Engine. In other cases, you can configure the client application to pass all files to Symantec Scan Engine. Then configure Symantec Scan Engine to scan those file types that are likely to contain viruses.

You must configure the client application to communicate with Symantec Scan Engine and to handle the results that are returned from Symantec Scan Engine.

How the application is configured to handle the results that are returned from Symantec Scan Engine can also depend on the capabilities of the third-party application, which includes, but is not limited to, the following:

- Blocking access to infected files or files that violate other configured policies
- Quarantining unrepairable files

For example, for content scanning, Symantec Scan Engine returns only the lookup results when you use audit mode. The client application applies the blocking policy based on the results. You can obtain information about configuring the client application to work with Symantec Scan Engine in audit mode by contacting Symantec Service and Support.

About configuring Symantec Scan Engine

Table 2-1 describes the minimum settings that you must configure for Symantec Scan Engine to perform scanning services.

Table 2-1 Symantec Scan Engine configuration options

Setting	Description
Select the ICAP protocol and configure protocol options	You must configure Symantec Scan Engine to use ICAP to communicate with clients that are running the proprietary version 1.0 of ICAP (RFC 3507, April 2003). Any appropriate client can use ICAP to communicate with Symantec Scan Engine to request scanning and repairing of files. You can configure multiple client applications that use different versions of ICAP to pass files to a single scan engine. If you select ICAP as the protocol to be used by Symantec Scan Engine, you must configure several ICAP-specific options. You must also configure the ICAP client to work with Symantec Scan Engine.

Table 2-1 Symantec Scan Engine configuration options *(continued)*

Setting	Description
Configure antivirus settings	You can configure certain aspects of antivirus scanning, including the following options: ■ Adjusting Bloodhound sensitivity ■ Specifying file types to scan ■ Establishing a mail filter policy
Specify processing limits	You can impose restrictions on the amount of resources that are used to handle individual files. These processing limits let you manage resources and protect your network against denial-of-service attacks.
Configure content filtering settings	You can configure content-filtering settings, which includes the following options: ■ Specifying content categories to deny ■ Specifying URLs to allow ■ Auditing URL sites rather than block access

For more information about how to configure these options, see the *Symantec Scan Engine Implementation Guide*.

Changing the ICAP response

In its default configuration, when ICAP is the communication protocol, Symantec Scan Engine sends a 201 Created response in the following scenarios:

- A virus was detected, but the file could not be repaired.
- A virus was detected and Symantec Scan Engine is configured for scan-only mode.

If a virus is detected and the file cannot be repaired or Symantec Scan Engine is configured for scan-only mode, Symantec Scan Engine includes a replacement file in the body of the response message. The replacement file is configurable through the Symantec Scan Engine administrative interface. The message informs users that access to a file is being denied because it contains a virus or content violation.

Symantec Scan Engine default behavior deviates from the ICAP 1.0 standard, which does not support automatically sending a replacement file. In the ICAP 1.0 standard, this type of context-sensitive behavior is performed by the client rather than by Symantec Scan Engine.

If your client application closely follows the ICAP 1.0 standard, you might need to change Symantec Scan Engine default ICAP response setting to receive an ICAP 403 response instead of a replacement file. Symantec Scan Engine sends a 403 response if the file is denied based on Symantec Scan Engine policy setting. If the file is acceptable, Symantec Scan Engine returns a 200 OK response. You should change the default ICAP response setting only if you are sure that the client application supports this behavior.

To make this change, you must edit the configuration.xml file using the XML modifier tool. The XML modifier tool is provided on the Symantec Scan Engine product CD. This tool is automatically installed when you install the product.

For more information, see the *Symantec Scan Engine Implementation Guide*.

To change the ICAP response

1. In the XML modifier tool, at the command line, type the following command:

   ```
   java -jar  xmlmodifier.jar
   ```

2. Type the following XPath:

   ```
   /configuration/protocol/ICAP/ICAPResponse/@value
   ```

   where @value is any of the following:

0	Send an ICAP 403 response
1	Send a replacement file. This is the default setting.

3. Stop and restart Symantec Scan Engine.

Chapter 3

Constructing clients using ICAP 1.0

This chapter includes the following topics:

How ICAP works

The Internet Content Adaptation Protocol (ICAP) is a request/response-based protocol that lets ICAP clients pass messages to ICAP servers for processing or adaptation. The client initiates the session by sending request messages over a TCP/IP connection to a passively waiting ICAP server on a designated port. (Port 1344 is the default ICAP port.)

The server then does the following:

- Runs the service that was requested, such as antivirus scanning
- Performs any transformations that are necessary, such as repairing an infected file
- Sends a response back to the client with any modified data

A single transport can be used for multiple request/response pairs. Requests are matched with responses by allowing only one outstanding request on a connection at a time. Multiple connections can be used.

About ICAP messages

ICAP clients and servers communicate through messages, which are similar in format to HTTP. ICAP messages consist of client requests and server responses. All ICAP messages consist of a start line, which includes a client request or server response (depending on the type of message), header fields, and the message body. A blank line precedes the message body to distinguish the headers from the message body.

Multiple HTTP message sections can be encapsulated in a single ICAP message for vectoring of requests, responses, and request/response pairs on an ICAP server.

Encapsulated messages must include an encapsulated header, which offsets the start of each encapsulated section from the start of the message body.

See "About encapsulated messages" on page 34.

Although request and response messages have unique headers, some headers are common to both requests and responses.

Table 3-1 lists the request/response headers that Symantec Scan Engine uses.

Table 3-1 Request/response headers

Header	Description
Connection	Specifies options that the message sender wants to use only for that connection and not for proxies over other connections. For example: Connection: close
Date	Provides the date and time that the message was created using standard HTTP date and time format. For example: Date: Tue, 5 July 2005 14:29:31 GMT

About ICAP requests

All ICAP client requests must start with a request line that includes the following components:

Method	ICAP command or operation to perform (for example, REQMOD)
Uniform Resource Identifier (URI)	Complete host name of the ICAP server and the path of the resource that is being requested
ICAP version	Version string for the current version of ICAP using the format ICAP/version number (for example ICAP/1.0)

The URI consists of the following components:

```
ICAP_URI = Scheme ":" Net_Path [ "?" Query ]
Scheme = "icap"
Net_Path = "//" Authority [ Abs_Path ]
Authority = [ userinfo "@" ] host [ ":" port ]
```

The request line specifies the ICAP resource that is being requested. Header fields follow with information, such as cache control and preview size. The header fields end with a blank line followed by the message body. The message body contains the encapsulated HTTP message sections that are being sent for scanning and modification.

Table 3-2 lists the request headers that are allowed in ICAP requests.

Table 3-2 Request headers

Header	Description
Allow	Lists the methods that the resource supports. For example, a client request can include an Allow: 204 header, which indicates that it will allow the server to reply to the message with a 204 No Content response if the file does not need modification. The client must buffer the message.
From	Provides the Internet email address for the user who is sending the client request. The address should use the standard HTTP mailbox format. For example: From: username@symantecdomain.com
Host	Specifies the host name and port number of the resource being requested.
Referer	Specifies the path that the client followed to obtain the URI. This optional header lets the server generate lists of backwards navigation links to resources and trace invalid links.

Table 3-2 Request headers *(continued)*

Header	Description
User-Agent	Identifies the software program that is used by the client that originated the request. This information is used for statistical purposes, to trace protocol violations, and to tailor responses to the software capabilities.
Preview (ICAP-specific header)	Lets the client send a portion of a file to Symantec Scan Engine for scanning. The client uses this header to specify the amount of data, in bytes, that will be sent for preview.
Encapsulated	Lists offsets of the start of each encapsulated section from the start of the message body. Opt-body=0 indicates filtering categories will be returned. See "About encapsulated messages" on page 34.
X-Filepath	Specifies the full file path to a local file.
X-URL-Blocked-Domain	Specifies the domain name for a URL request that Symantec Scan Engine has blocked. This header is an optional field for a URL filtering request. This header is sent only when a URL is blocked at the domain level. For example, if Symantec Scan Engine is configured to block uninvitedads.com, then all URL scanning requests from uninvitedads.com domain receive this header in the ICAP response. URL filtering requests are identified by following ICAP services: ■ SYMCScanReq-URL ■ SYMCScanReq-AV-URL

About ICAP methods

Symantec Scan Engine supports the following ICAP methods:

OPTIONS (options mode) — Lets the client obtain information from an ICAP server about available services.

See "How to determine which services are supported (OPTIONS)" on page 41.

REQMOD (request modification mode)	Lets the client send URLs to Symantec Scan Engine for scanning services. In request modification mode, the ICAP client receives a request from a user, usually a Web-browser. The client passes the data to Symantec Scan Engine for evaluation and processing. See "Scanning HTTP requests (REQMOD)" on page 55.
RESPMOD (response modification mode)	Lets the client send files to Symantec Scan Engine for scanning services. In response modification mode, the ICAP client receives a data response from an origin server. The client passes the data to Symantec Scan Engine for evaluation and post-processing. See "Scanning HTTP responses (RESPMOD)" on page 59.
FILEMOD (file modification mode)	Lets the client pass a file name and path to Symantec Scan Engine so that Symantec Scan Engine can scan the file in place (rather than streaming the file to Symantec Scan Engine for scanning). **Note:** File modification mode deviates from the ICAP 1.0 specification that is presented in RFC 3507 (April 2003). See "Scanning non-HTTP data (RESPMOD and FILEMOD)" on page 63.

About ICAP services

Symantec Scan Engine supports antivirus and URL content-filtering scanning. You must include an ICAP service name in the ICAP URI to specify the type of scanning that you want Symantec Scan Engine to perform. Previous versions of Symantec AntiVirus Scan Engine used a different ICAP service naming convention. These services are supported for backward compatibility. New clients should use the new services, which include enhancements over the previous services.

Table 3-3 describes the ICAP services, descriptions, and the supported methods.

Table 3-3 ICAP services

ICAP service name	Description	Method options	Legacy or New service
AVSCANREQ	Lets you apply antivirus scanning to client request data (compatible with Symantec AntiVirus Scan Engine)	REQMOD	Legacy
AVSCANRESP	Lets you apply antivirus scanning to client response data (compatible with Symantec AntiVirus Scan Engine)	RESPMOD, FILEMOD	Legacy
AVSCAN	Lets you apply antivirus scanning to client response data	RESPMOD, FILEMOD	Legacy
SYMCScanReq-AV	HTTP request is provided to Symantec Scan Engine for antivirus scanning	REQMOD	New
SYMCScanReq-AV-URL	HTTP request is provided to Symantec Scan Engine for antivirus and URL-filtering scanning	REQMOD	New
SYMCScanReq-URL	HTTP request is provided to Symantec Scan Engine for URL-filtering scanning	REQMOD	New
SYMCScanResp-AV	HTTP response is provided to Symantec Scan Engine for antivirus scanning	RESPMOD, FILEMOD	New
SYMCScanResp-AV-DDR	HTTP response is provided to Symantec Scan Engine for antivirus and URL-filtering scanning	RESPMOD, FILEMOD	New
SYMCScanResp-DDR	HTTP response is provided to Symantec Scan Engine for DDR scanning	RESPMOD, FILEMOD	New

The ICAP service argument is used to specify the antivirus scanning policy. The action=repairpolicy argument can override the antivirus scanning repair policy for Symantec Scan Engine. For services that do not perform antivirus scanning,

SYMCScanReq-URL and SYMCScanResp-DDR, the argument is ignored. The ICAP service argument is as follows:

action=repairpolicy

where repair policy consists of any of the following:

- scanrepairdelete
- scanrepair
- scandelete
- scan

You can append the argument to the service name by adding a question mark and then the argument. For example:

```
SYMCSCANRESP-AV?action=scanrepairdelete
```

About ICAP responses

ICAP client responses start with a status line, which includes the ICAP version and a status or response code. For example:

```
ICAP/1.0 200 OK
```

Table 3-4 lists the response codes that Symantec Scan Engine uses (response codes vary depending on the type of request).

Table 3-4 Status codes

Status code	Text	Description
100	Continue	Symantec Scan Engine completed a preview and requires additional data.
200	OK	The server processed the request successfully.
201	Created	A violation was detected, and the file has been repaired or replaced. The response also includes the modified data.
204	No content necessary	Scanning is not required, and the client sent an Allow: 204 header, which indicates that Symantec Scan Engine does not need to return data to the client.
400	Bad request	There was a syntax error or other problem parsing the request.

Table 3-4 Status codes *(continued)*

Status code	Text	Description
403	Forbidden not repaired	The data contained a scanning violation and cannot be repaired, or Symantec Scan Engine is configured for scan-only mode. **Note:** This response is the standard ICAP behavior, as documented in the ICAP 1.0 specification. Symantec Scan Engine does not follow this behavior by default. If your client application follows the ICAP 1.0 standard, you might need to change the default setting for an ICAP response.
404	Not found	The URI in the request does not correspond to an available service.
500	Internal server error	There is a generic problem with the server.
502	Bad gateway	The server cannot access a file at the specified location (X-Filepath:).
505	Version not supported	Only ICAP 1.0 is supported with this method.
506	Server too busy.	Lets the client application know that the Symantec Scan Engine has reached its threshold for scanning requests and that the server is too busy to process the request.
551	Resource unavailable	The server has a memory or disk problem.
558	Aborted no scanning license	Symantec Scan Engine is unable to scan the data because a valid license does not exist. See "About licensing" on page 14.

The status line is followed by one or more response headers that let the server pass additional information (for example, information that cannot be placed in the status line) to the client.

Table 3-5 lists the response headers that Symantec Scan Engine uses (response headers vary depending on the type of request).

Table 3-5 Response headers

Header	Description
Date	Specifies the date and time as set on the server clock.

Table 3-5 Response headers *(continued)*

Header	Description
Service	Specifies the name and version number of the ICAP server.
Server-ID	Specifies the service requested.
ISTag (ICAP service tag)	Lets an ICAP server send service-specific information to an ICAP client. This data can be used to validate whether a server response, including cached data, is still valid. Symantec Scan Engine returns an ISTag with every response to indicate the state of the service, including all relevant configuration options and scanning definitions. Cached data that does not match the current ISTag is no longer valid. The value is a 32-character hexadecimal string. For example: ISTag: "B3C20CFCACEDA72CF16F6AEC119B2981"
Methods	Specifies the methods that are supported by the service that you queried.
Allow	Lists the optional ICAP features that the server supports.
Preview	Indicates the number of bytes of data that should be sent to Symantec Scan Engine for preview.
Transfer-Preview	Lists the file extensions that should be sent to Symantec Scan Engine for preview before sending the entire file. An asterisk (*) wildcard character represents the default behavior for all file extensions that are not specified in another transfer-type header.
Transfer-Complete	Lists the file extensions that should always be sent in their entirety to Symantec Scan Engine and that should not be previewed. An asterisk (*) wildcard character represents the default behavior for all file extensions that are not specified in another Transfer-type header.
Max-Connections	Indicates the maximum number of simultaneous ICAP connections that the server supports.
Options-TTL	Indicates the time (in seconds) during which the response is valid or cached. A blank header indicates that the response does not expire.

Table 3-5 Response headers *(continued)*

Header	Description
Encapsulated	Lists offsets of the start of each encapsulated section from the start of the message body.
X-AV-License	Indicates whether a valid antivirus scanning license is installed on Symantec Scan Engine, where 1 indicates a valid antivirus scanning license and 0 indicates no valid antivirus scanning license. The X-AV-License header does not include whether a content-filtering license is installed or valid.
X-Allow-Out	Indicates the custom X-headers that are returned in responses from Symantec Scan Engine.
X-Definition-Info	Indicates the date and revision number of the virus definitions in the following format: YYYYMMDD.RRR where YYYY is the four-digit year, MM is the month, DD is the day, and RRR is the revision number.
Opt-Body-Type	Indicates opt-body will be an attribute list. The attribute list begins with the following: X-ICAP-Attribute-<serviceid> This tag indicates that the start of the category list begins after this CRLF delimited header. The list contains the valid categories for the requested Service-ID. A semi-colon (;) followed by a blank line indicates the end of the list.
X-Outer-Container-Is-Mime	(Optional header) An integer value that indicates whether the outer container is a valid MIME container. A client application can use this information to reject content that does not meet this criteria. Zero (0) indicates that the outer container is not a valid MIME container, and one (1) indicates that the outer container is a valid MIME container.

Table 3-5 Response headers *(continued)*

Header	Description
X-Infection-Found	Provides information about an infection that is found. Only one violation is reported, regardless of the number of violations found. The header provides the following information regarding the infection: ■ Violation type An integer value for the violation. Zero (0) indicates a virus, one (1) indicates a mail policy violation, and two (2) indicates a container violation or malformity. ■ Resolution An integer value that indicates what action was taken on the file. Zero (0) indicates that the file was not fixed, one (1) indicates that the file was repaired, and two (2) indicates that access to the file was blocked. ■ Threat value String that describes the virus or violation that was found. This header applies to legacy services only. See "About ICAP services" on page 27.

Table 3-5 Response headers *(continued)*

Header	Description
X-Violations-Found	Indicates the total number of violations (either an infection or a policy violation) that were found in the scanned data. If violations were detected, the header is followed by a series of indented lines that provide the following information for each violation: ■ File name The name of the scanned file or the name of a nested component within the scanned file. Each component name is separated by a forward slash mark (/). ■ Violation name The English-readable name of the violation. ■ Violation ID A numeric code for the violation. ■ Disposition An integer value that indicates what action was taken to fix the file. Zero (0) indicates that the file was not fixed, one (1) indicates that the file was repaired, and two (2) indicates that the file was deleted.
X-URL-Blocked-Domain	Specifies the domain name for a URL request that Symantec Scan Engine has blocked. This header is an optional field for a URL filtering request. This header is sent only when a URL is blocked at the domain level. For example, if Symantec Scan Engine is configured to block uninvitedads.com, then all URL scanning requests from uninvitedads.com domain receive this header in the ICAP response. URL filtering requests are identified by following ICAP services: ■ SYMCScanReq-URL ■ SYMCScanReq-AV-URL

About encapsulated messages

The ICAP encapsulation model provides a lightweight means of packaging multiple HTTP message sections into a single ICAP message for vectoring of requests,

responses, and request/response pairs on an ICAP server. An encapsulated section can consist of HTTP message headers and bodies.

Encapsulated HTTP message bodies must be transferred using chunked transfer encoding. This keeps the transport-layer connection between the client and server open for later use and lets the server send incremental responses to reduce the latency that is perceived by users. Encapsulated HTTP headers are not chunked. This lets the ICAP client copy the header directly from the HTTP client to the ICAP server without having to reprocess it.

Note: The chunked transfer encoding modifies the body of a message so that it can be transferred as a series of chunks, each with its own (hexadecimal) size indicator, followed by an optional footer that contains entity-header fields. For more information, see the HTTP/1.1 specification (RFC 2616, section 3.6.1).

The encapsulated header must be included in every ICAP message, except for OPTIONS requests. This header provides information about where each encapsulated section and message body starts and ends.

For example:

```
Encapsulated: req-hdr=0, res-hdr=45, res-body=100
```

This example indicates that the message encapsulates a group of request headers, response headers, and a response body at 0, 45, and 100 byte offsets. Byte offsets use a decimal format. Chunk sizes within an encapsulated body use a hexadecimal format. If no message body is sent, a null-body entity is used.

Encapsulated headers use the following syntax:

```
encapsulated_header: "Encapsulated: " encapsulated_list
encapsulated_list: encapsulated_entity |
      encapsulated_entity ", " encapsulated_list
encapsulated_entity: reqhdr | reshdr | reqbody | resbody | optbody
reqhdr  = "req-hdr" "=" (decimal integer)
reshdr  = "res-hdr" "=" (decimal integer)
reqbody = { "req-body" | "null-body" } "=" (decimal integer)
resbody = { "res-body" | "null-body" } "=" (decimal integer)
optbody = { "opt-body" | "null-body" } "=" (decimal integer)
```

Encapsulated headers must end with a blank line to make them readable and to terminate line-by-line HTTP parsers.

Where to find more information on ICAP

Symantec Scan Engine supports the ICAP 1.0 specification that is presented in RFC 3507 (April 2003). Much of the information in the *Symantec Scan Engine Software Developer's Guide* is obtained directly from the specification. However, the specification contains more extensive examples and additional information. Developers are encouraged to consult the specification, as well as other sources.

For more information about ICAP specifications, go the following URL:

www.i-cap.org

The content filtering feature was implemented using the ICAP Extensions Internet Draft, section 5.2.

About the scanning process

Symantec Scan Engine provides antivirus and URL-filtering scanning services to ICAP clients.

The scanning services are provided for non-HTTP scanning, such as email and FTP traffic, and the following user scenarios:

- A user requests a file from a Web site
- A user posts a file to a Web site
- A user receives a file from a Web site

When an ICAP client receives an HTTP request, it is encapsulated into an ICAP REQMOD request and sent to a scan engine ICAP service. The service that is specified in the URI must support the REQMOD method. This is a simple request with HTTP request headers and no body. Because there is no data to scan for viruses, there is no value added by an antivirus-only service like SYMCScanReq-AV. The other REQMOD services, SYMCScanReq-URL and SYMCScanReq-AV-URL, will do the same.

If an HTTP request contains a file, such as a posting to a Web site, there is value to antivirus scanning. Symantec Scan Engine can either repair the file (return a modified HTTP request) or notify the user (return an HTTP response), but not both. Therefore, for REQMOD antivirus scanning, the file is never repaired (scanpolicy = scan). The user receives a Web page that states that the post failed because the file was infected or triggered another violation.

When an ICAP client receives an HTTP response from an origin server, it is encapsulated into an ICAP RESPMOD request and sent to Symantec Scan Engine. The service specified must support RESPMOD. The service specified determines the kind of scanning that is performed on the HTTP body: antivirus, URL content filtering, or both. Because Symantec Scan Engine has an option to ignore the URL

content-filtering results based on the URL, Symantec Scan Engine must also scan the URL in the HTTP request headers.

The ICAP RFC specifies that these headers are optional for a response modification request, but for clients that use Symantec Scan Engine, they are required. It is possible to get URL filtering on a RESPMOD request, but it is more efficient to block URLs during the request before expending resources to retrieve the data.

Symantec Scan Engine can be used to scan non-HTTP data, such as files on disk, email messages, and FTP traffic. You can scan non-HTTP data by creating an ICAP RESPMOD request with a minimal set of fabricated HTTP request headers and HTTP response headers. Typically the HTTP request headers include an HTTP request line that contains the file name (Symantec Scan Engine provides features based on file name, so this is required) and a Host header. The HTTP response headers contain an HTTP response line: HTTP/1.1 200 OK.

Symantec Scan Engine also supports an extension to ICAP (FILEMOD), which lets the client send a request to have a file scanned on-disk to avoid sending the file across the network. An ICAP client requests on-disk scanning by sending an ICAP FILEMOD request to Symantec Scan Engine. This request is composed of ICAP headers only and no ICAP body (Encapsulated: null-body=0). Included in the headers is the path that Symantec Scan Engine can use to access the file. The ICAP response will be similar to a network scan, except that no data will be returned. Any modification is done on the actual file.

Before sending ICAP requests, the client can query the ICAP server by using the OPTIONS method to determine which services are supported.

See "How to determine which services are supported (OPTIONS)" on page 41.

About scanning URLs

When scanning URLs, the URL can be constructed in the following ways:

When the HTTP request line contains only the path information, it is combined with the host name that is provided in the Host header to create the full URL.	GET/directory/file.exe HTTP/1.1 Host: client.symantecdomain.com The URL is client.symantecdomain.com/directory/file.exe
The URL can also be sent completely in the HTTP request line. This is typically the case for HTTP POST requests.	POST http://client.symantecdomain.com/directory/file.exe HTTP/1.1 Host: client.symantecdomain.com The URL is extracted from the request line and the Host header is ignored.

About sending files for scanning

Symantec Scan Engine supports multiple URIs for scanning services. The URI uses the following format:

```
icap://server.name:port/servicename
```

where server.name is the name of the server on which Symantec Scan Engine is running. The port number is optional if Symantec Scan Engine is running on port 1344, which is the default ICAP port. Servicename is one of the ICAP services.

See "About ICAP services" on page 27.

About sending portions of files for preview

Symantec Scan Engine can preview data to determine whether it needs to be scanned based on known virus behavior. For example, .gif files are typically not scanned because they generally do not contain executable code. The rules for which types of files are suitable for preview are determined by the exclusion lists that are configured in the Symantec Scan Engine administrative interface.

Before a file is sent for scanning, the client should send an OPTIONS request to determine whether a file type is suitable for preview and how much data should be sent.

Symantec Scan Engine provides this information in the following headers of the OPTIONS response message:

Preview	Indicates the preferred number of bytes of data that can be sent
Transfer-Complete	Indicates which file types should be sent in their entirety
Transfer-Preview	Indicates which file types should be sent for preview

See "How to determine which services are supported (OPTIONS)" on page 41.

Table 3-6 details Symantec Scan Engine scanning behavior that is based on the scanning policies that you configure.

Table 3-6 Scanning behavior

Scanning policy	Transfer-Complete header	Transfer-Preview header	Scanning behavior
Previews all files	Not used	Asterisk (*) character	All files are previewed for unwanted content.

Table 3-6 Scanning behavior *(continued)*

Scanning policy	Transfer-Complete header	Transfer-Preview header	Scanning behavior
Scan all files regardless of extension	Asterisk (*) character	Not used	Symantec Scan Engine scans every file in its entirety without previewing it first.
Scan all files except those with the following extensions (exclusion list)	Asterisk (*) character	List of file extensions	Symantec Scan Engine previews the file types that are listed in the Transfer-Preview header for unwanted content. All other file types, including unidentified file types, are scanned in their entirety.

For more information, see the *Symantec Scan Engine Implementation Guide*.

If an OPTIONS response indicates that a file is suitable for preview, the client should include a Preview header in the request message that indicates the portion of data, in bytes, that is being sent for preview. Symantec Scan Engine evaluates the initial chunk of data to determine whether a full scan is required. If so, Symantec Scan Engine requests the remainder of the data. Scan results are returned in the RESPMOD response message.

About allowing No Content responses

The Allow: 204 header is an optional request header that lets Symantec Scan Engine return a 204 No Content response code if the message does not require modification. This can optimize server performance because Symantec Scan Engine can determine whether a file is uninfected without having to receive the entire message, and Symantec Scan Engine does not have to return a message. The processing burden is placed on the client, which must buffer the entire message during the scan. Symantec Scan Engine returns a 204 No Content response outside of a preview only if the client request includes an Allow: 204 header.

About non-viral threat category responses

When Symantec Scan Engine detects a non-viral threat during a scan, the ICAP X-Violations-Found response header includes the threat category name (NonViralThreat) in the ThreatDescription field. The threat category name is

appended to the virus name with a delimiter pipe. For example, ThreatDescription = <VirusName> | NonViralThreat=<CategoryName>.

The following is a list of all the values currently supported for categories for non-viral threats:

- Adware
- Spyware
- Reserved Malicious
- Malicious
- Heuristic
- Hack Tools
- Trackware
- Dialers
- Joke Programs
- Remote Access
- Security Risks

By default, Symantec Scan Engine does not send the threat category name in the header. To configure Symantec Scan Engine to send the threat category name, you must change the EnableNonViralThreatCategoryResp value in the configuration.xml file to true. For more information about how to modify the configuration.xml file using the XML modifier command-line tool, see the *Symantec Scan Engine Implementation Guide*.

An example of an X-Violations-Found response header with a NonViralTreat category is as follows:

```
C: RESPMOD icap://127.0.0.1:1344/SYMCScanResp-AV ICAP/1.0
C: Host: 127.0.0.1:1344
C: Connection: close
C: Encapsulated: req-hdr=0, res-hdr=44, res-body=63
C:
C: get / HTTP/1.1
C: Host:icheck.symantec.com
C:
C: HTTP/1.1 200 OK
C:

S: ICAP/1.0 201 Created
ISTag: "C5D8B6B34BC2785CE5017A242480E9C2"
```

```
Date: Mon Jan 21 05:10:32 2008 GMT
Service: Symantec Scan Engine/5.2.0.12
Service-ID: SYMCSCANRESP-AV
X-Violations-Found: 1
 index.html
 Adware.Unknown| NonViralThreat=Adware
 4294909841
 2
X-Outer-Container-Is-Mime: 0
Encapsulated: res-hdr=0, res-body=83
```

See "About ICAP responses" on page 29.

How to determine which services are supported (OPTIONS)

The OPTIONS method lets a client application query an ICAP server for information about supported services and commands and preferred file handling methods. The client application should perform this query before sending files for scanning.

The OPTIONS method consists of a request line that contains the URI for the Symantec Scan Engine service that you want to query.

When Symantec Scan Engine receives an OPTIONS request from a client application, it sends a response that includes the following information:

- Maximum number of simultaneous connections allowed
- Preferred data preview size
- Preferred file handling methods
- Supported methods (REQMOD, RESPMOD, FILEMOD)
- For content-filtering services (-URL, -DDR), the list of content categories that are available is returned in an encapsulated opt-body section
  See "About licensing" on page 14.

See "About ICAP responses" on page 29.

About using the OPTIONS request to determine if the server is overloaded

If you use ICAP, the ICAP threshold client notification feature is enabled by default. When the number of queued requests for a Symantec Scan Engine exceeds its threshold, Symantec Scan Engine rejects the scan request. It notifies the client

that the server has reached the queued request threshold. The client can then adjust the load balancing, which prevents the server from being overloaded with scan requests. This feature lets the client applications that pass files to Symantec Scan Engine benefit from load-balanced scanning without any additional effort.

You can use the OPTIONS request to determine if Symantec Scan Engine is overloaded. Send an OPTIONS request to the Symantec Scan Engine. If Symantec Scan Engine is not busy, it will send the standard OPTIONS response to the connector and will keep the connection open. If Symantec Scan Engine is too busy to process the request, it will reply with 506 response “Server too busy” and will close the connection. In this case, the load balancing decision should be made by the connector.

A sample ICAP response body for code 506 is as follows:

```
ICAP/1.0 506 Server too busy
ISTag: "638FC09D7F7CCFD3DCF0E659FFEAD53F"
Date: Tue Jun 26 05:52:14 2007 GMT
Service: Symantec Scan Engine/5.2.0.1
Service-ID: Respmod AV Scan
```

About querying antivirus-only services

The antivirus-only services include the following:

AVSCAN	Provides antivirus scanning for HTTP requests. The request must include a file (uploading) and use the HTTP POST method. Otherwise, an ICAP 200 OK response is returned. This is a legacy service and should not be used by new clients.
AVSCANREQ	Provides antivirus scanning for HTTP requests. The request must include a file (uploading) and use the HTTP POST method. Otherwise, an ICAP 200 OK response is returned. This is a legacy service and should not be used by new clients.
AVSCANRESP	Provides antivirus scanning for HTTP requests. The request must include a file (uploading) and use the HTTP POST method. Otherwise, an ICAP 200 OK response is returned. This is a legacy service and should not be used by new clients.
SYMScanReq-AV	Provides antivirus scanning for HTTP requests. The request must include a file (uploading) and use the HTTP POST method. Otherwise, an ICAP 200 OK response is returned.

SYMCScanResp-AV	Provides antivirus scanning for content that is downloaded from the Internet.

An OPTIONS request to an antivirus-only service provides the validity of the antivirus license, the antivirus definitions version, and an ISTag, which indicates the state of the configuration settings.

See "About licensing" on page 14.

Table 3-3 lists the services that you can use for scan, repair, and delete functions. You specify the scanning preferences through the Symantec Scan Engine administrative user interface. You must have a valid license installed to perform scanning functions.

Examples of OPTIONS requests for antivirus scanning are as follows:

```
/* AV is only performed on POST requests */
OPTIONS icap://icapclient.sse.com/symcscanreq-url ICAP/1.0
Host: icapclient.sse.com


OPTIONS icap://icapclient.sse.com/symcscanresp-ddr ICAP/1.0
Host: icapclient.sse.com
```

About querying content-filtering services

The content-filtering services include the following:

SYMCScanReq-URL	Blocks requests to sites based on the URL. This includes comparing the requested URL to Symantec's URL category lists and optionally to use Dynamic Document Review (DDR) technology to infer the content of the site based on the words that are used in the URL.
SYMCScanResp-DDR	Used to categorize and possibly block content that is retrieved from a site by using DDR technology. A list of categories that the content can be identified as belonging to is returned. This service will also examine the URL if the optional HTTP request headers are included in the ICAP request. To enhance performance, URLs should be scanned by the SYMCScanReq-URL service before retrieving the content from the origin server.

For more information about how URL and DDR filtering work, see the *Symantec Scan Engine Implementation Guide*.

Examples of OPTIONS requests for URL filtering are as follows:

```
OPTIONS icap://sse.com/symcscanreq-url ICAP/1.0
Host: icapclient.sse.com

OPTIONS icap://sse.com/symcscanresp-ddr ICAP/1.0
Host: icapclient.sse.com
```

About querying antivirus and content-filtering services

The antivirus and content filtering scanning services include the following:

SYMCScanReq-AV-URL	Performs content filtering on HTTP requests and scans HTTP bodies (files) for malicious content.
SYMCScanResp-AV-DDR	Combines content filtering and antivirus scanning.

Examples of OPTIONS requests for antivirus and content filtering are as follows:

```
/* AV is only performed on POST requests */
OPTIONS icap://sse.com/symcscanreq-av-url ICAP/1.0
Host: icapclient.sse.com

OPTIONS icap://sse.com/symcscanresp-av-ddr ICAP/1.0
Host: icapclient.sse.com
```

OPTIONS examples

Examples of OPTIONS services are as follows:

- OPTIONS antivirus-scanning example
- OPTIONS content-filtering scanning example
- OPTIONS antivirus and content-filtering scanning example

OPTIONS antivirus-scanning example

The antivirus scanning services perform scanning, repair, and delete functions by using the scanning preferences that you specify through the Symantec Scan Engine administrative user interface. Use the following format to specify the URI:

```
icap://<Server>/<service>
```

A sample OPTIONS request on SYMCScanReq-AV is as follows:

```
C: OPTIONS icap://172.16.11.19:1344/SYMCScanReq-AV ICAP/1.0
C: Host: 172.16.11.19:1344
C: Connection: close
C: Encapsulated: null-body=0
C:
S: ICAP/1.0 200 OK
S: Date: Mon Jun 27 11:50:14 2005 GMT
S: Methods: REQMOD
S: Service: Symantec Scan Engine/5.0.0.23
S: Service-ID: SYMCSCANREQ-AV
S: ISTag: "B3C20CFCACEDA72CF16F6AEC119B2981"
S: X-Definition-Info: 20050622.017
S: Max-Connections: 128
S: X-Allow-Out: X-Outer-Container-Is-Mime, X-Infection-Found
S: X-Allow-Out: X-Definition-Info, X-AV-License
S: X-Allow-Out: X-Violations-Found
S: Allow: 204
S: Options-TTL: 3600
S: Preview: 4
S: Transfer-Preview: *
S: X-AV-License: 1
S: Encapsulated: null-body=0
S:
```

This response informs the client of the following:

- Besides the OPTIONS method, the only supported method is REQMOD.
- The optional 204 shortcut is supported.
- Four bytes of preview information are preferred.
- All files should be sent for preview.
- The data in the response can be cached up to one hour.
- The server supports a maximum of 128 simultaneous connections.
  The maximum number of simultaneous connections that Symantec Scan Engine supports varies on the operating environment.

A sample OPTIONS response on SYMCScanResp-AV is as follows:

```
C: OPTIONS icap://172.16.11.19:1344/SYMCScanResp-AV ICAP/1.0
C: Host: 172.16.11.19:1344
C: Connection: close
C: Encapsulated: null-body=0
```

```
C:
S: ICAP/1.0 200 OK
S: Date: Mon Jun 27 11:50:34 2005 GMT
S: Methods: RESPMOD, FILEMOD
S: Service: Symantec Scan Engine/5.0.0.23
S: Service-ID: SYMCSCANRESP-AV
S: ISTag: "B3C20CFCACEDA72CF16F6AEC119B2981"
S: X-Definition-Info: 20050622.017
S: Max-Connections: 128
S: X-Allow-Out: X-Outer-Container-Is-Mime, X-Infection-Found
S: X-Allow-Out: X-Definition-Info, X-AV-License
S: X-Allow-Out: X-Violations-Found
S: Allow: 204
S: Options-TTL: 3600
S: Preview: 4
S: Transfer-Preview: *
S: X-AV-License: 1
S: Encapsulated: null-body=0
S:
```

This response informs the client of the following:

- Besides the OPTIONS method, the only supported methods are RESPMOD and FILEMOD.
- The optional 204 shortcut is supported.
- Four bytes of preview information are preferred.
- All files should be sent for preview.
- The data in this response can be cached up to one hour.
- The server supports a maximum of 128 simultaneous connections.
  The maximum number of simultaneous connections that Symantec Scan Engine supports varies on the operating environment.

OPTIONS content-filtering scanning example

The content-filtering scanning services perform content filtering based on the options that you specify through the Symantec Scan Engine user interface.

Use the following format to specify the URI:

```
icap://<Server>/<service>
```

A sample OPTIONS request on SYMCScanReq -URL is as follows:

```
C: OPTIONS icap://172.16.11.19:1344/SYMCScanReq-URL ICAP/1.0
C: Host: 172.16.11.19:1344
C: Connection: close
C: Encapsulated: null-body=0
C:
S: ICAP/1.0 200 OK
S: Date: Mon Jun 27 11:50:28 2005 GMT
S: Methods: REQMOD
S: Service: Symantec Scan Engine/5.0.0.23
S: Service-ID: SYMCSCANREQ-URL
S: ISTag: "91D1448B9843D9D5CB0850736024EE90"
S: X-Definition-Info: 20050622.017
S: Max-Connections: 128
S: X-Allow-Out: X-Outer-Container-Is-Mime, X-Infection-Found
S: X-Allow-Out: X-Definition-Info, X-AV-License
S: X-Allow-Out: X-Violations-Found
S: Allow: 204
S: Options-TTL: 3600
S: Preview: 4
S: Transfer-Preview: *
S: X-AV-License: 1
S: Encapsulated: opt-body=0
S: X-Allow-Out: X-Attribute
S: Opt-Body-Type: Attribute-List
S:
S: 20d
S: X-ICAP-Attribute-SYMCSCANREQ-URL
S: Adult Humor
S: Alcohol-Tobacco
S: Anonymous Proxies
S: Crime
S: Drugs/Advocacy
S: Drugs/Non-medical
S: Entertainment/Games
S: Entertainment/Sports
S: Finance
S: Gambling
S: Humor
S: Interactive/Chat
S: Interactive/Mail
S: Intolerance
S: Job Search
S: News
```

```
S: Occult/New Age
S: Prescription Medicine
S: Real Estate
S: Religion
S: Sex/Acts
S: Sex/Attire
S: Sex/Nudity
S: Sex/Personals
S: Sex Education/Advanced
S: Sex Education/Basic
S: Sex Education/Sexuality
S: Travel
S: Vehicles
S: Violence
S: Weapons
S: AllowURLsCategory
S: AllowURLsWithDDRCategory
S: ;
S:
S: 0
S:
```

This response informs the client of the following:

- Besides the OPTIONS method, the only supported method is REQMOD.
- The optional 204 shortcut is supported.
- Four bytes of preview information are preferred.
- All files should be sent for preview.
- The data in this response can be cached up to one hour.
- The server supports a maximum of 128 simultaneous connections.
  The maximum number of simultaneous connections that Symantec Scan Engine supports varies depending on the operating environment.

You will see new information that starts with Encapsulated: opt-body=0. When scanning for content, an opt-body header is returned that indicates a list of categories that are supported by Symantec Scan Engine. X-Allow-Out contains an addition of X-Attribute and the Opt-Body-Type is Attribute-List.

A sample OPTIONS response on SYMCScanResp-DDR is as follows:

```
C: OPTIONS icap://172.16.11.19:1344/SYMCScanResp-DDR ICAP/1.0
C: Host: 172.16.11.19:1344
C: Connection: close
```

```
C: Encapsulated: null-body=0
C:
S: ICAP/1.0 200 OK
S: Date: Mon Jun 27 11:50:43 2005 GMT
S: Methods: RESPMOD, FILEMOD
S: Service: Symantec Scan Engine/5.0.0.23
S: Service-ID: SYMCSCANRESP-DDR
S: ISTag: "E67C87BA7D981353864B63F6001E8D53"
S: X-Definition-Info: 20050622.017
S: Max-Connections: 128
S: X-Allow-Out: X-Outer-Container-Is-Mime, X-Infection-Found
S: X-Allow-Out: X-Definition-Info, X-AV-License
S: X-Allow-Out: X-Violations-Found
S: Allow: 204
S: Options-TTL: 3600
S: Preview: 4
S: Transfer-Preview: *
S: X-AV-License: 1
S: Encapsulated: opt-body=0
S: X-Allow-Out: X-Attribute
S: Opt-Body-Type: Attribute-List
S:
S: 20e
S: X-ICAP-Attribute-SYMCSCANRESP-DDR
S: Adult Humor
S: Alcohol-Tobacco
S: Anonymous Proxies
S: Crime
S: Drugs/Advocacy
S: Drugs/Non-medical
S: Entertainment/Games
S: Entertainment/Sports
S: Finance
S: Gambling
S: Humor
S: Interactive/Chat
S: Interactive/Mail
S: Intolerance
S: Job Search
S: News
S: Occult/New Age
S: Prescription Medicine
S: Real Estate
```

```
S: Religion
S: Sex/Acts
S: Sex/Attire
S: Sex/Nudity
S: Sex/Personals
S: Sex Education/Advanced
S: Sex Education/Basic
S: Sex Education/Sexuality
S: Travel
S: Vehicles
S: Violence
S: Weapons
S: AllowURLsCategory
S: AllowURLsWithDDRCategory
S: ;
S:
S: 0
S:
```

This response informs the client of the following:

- Besides the OPTIONS method, the only supported method is REQMOD.
- The optional 204 shortcut is supported.
- Four bytes of preview information are preferred.
- All files should be sent for preview.
- The data in this response can be cached up to one hour.
- The server supports a maximum of 128 simultaneous connections.
  The maximum number of simultaneous connections that Symantec Scan Engine supports varies on the operating environment.

You will see new information that starts with Encapsulated: opt-body=0. When scanning for content, an opt-body header is returned that indicates a list of categories that are supported by Symantec Scan Engine. X-Allow-Out contains an addition of X-Attribute and the Opt-Body-Type is Attribute-List.

OPTIONS antivirus and content-filtering scanning example

The SYMCScan* services perform antivirus scanning, repair, and delete functions and content filtering based on the scanning preferences that you specify through the Symantec Scan Engine administrative user interface.

Use the following format to specify the URI:

```
icap://<Server>/<service>
```

A sample OPTIONS request on SYMCScanReq-AV-URL is as follows:

```
C: OPTIONS icap://172.16.11.19:1344/SYMCScanReq-AV-URL ICAP/1.0
C: Host: 172.16.11.19:1344
C: Connection: close
C: Encapsulated: null-body=0
C:
S: ICAP/1.0 200 OK
S: Date: Mon Jun 27 11:50:23 2005 GMT
S: Methods: REQMOD
S: Service: Symantec Scan Engine/5.0.0.23
S: Service-ID: SYMCSCANREQ-AV-URL
S: ISTag: "B397E3F3013EFB2B1D3F214DD43B9D4E"
S: X-Definition-Info: 20050622.017
S: Max-Connections: 128
S: X-Allow-Out: X-Outer-Container-Is-Mime, X-Infection-Found
S: X-Allow-Out: X-Definition-Info, X-AV-License
S: X-Allow-Out: X-Violations-Found
S: Allow: 204
S: Options-TTL: 3600
S: Preview: 4
S: Transfer-Preview: *
S: X-AV-License: 1
S: Encapsulated: opt-body=0
S: X-Allow-Out: X-Attribute
S: Opt-Body-Type: Attribute-List
S:
S: 210
S: X-ICAP-Attribute-SYMCSCANREQ-AV-URL
S: Adult Humor
S: Alcohol-Tobacco
S: Anonymous Proxies
S: Crime
S: Drugs/Advocacy
S: Drugs/Non-medical
S: Entertainment/Games
S: Entertainment/Sports
S: Finance
S: Gambling
S: Humor
S: Interactive/Chat
S: Interactive/Mail
```

```
S: Intolerance
S: Job Search
S: News
S: Occult/New Age
S: Prescription Medicine
S: Real Estate
S: Religion
S: Sex/Acts
S: Sex/Attire
S: Sex/Nudity
S: Sex/Personals
S: Sex Education/Advanced
S: Sex Education/Basic
S: Sex Education/Sexuality
S: Travel
S: Vehicles
S: Violence
S: Weapons
S: AllowURLsCategory
S: AllowURLsWithDDRCategory
S: ;
S:
S: 0
S:
```

This response informs the client of the following:

- Besides the OPTIONS method, the only supported method is REQMOD.
- The optional 204 shortcut is supported.
- Four bytes of preview information are preferred.
- All files should be sent for preview.
- The data in this response can be cached up to one hour.
- The server supports a maximum of 128 simultaneous connections.
  The maximum number of simultaneous connections that Symantec Scan Engine supports varies on the operating environment.

You will see new information that starts with Encapsulated: opt-body=0. When scanning for content, an opt-body header is returned that indicates a list of categories that are supported by Symantec Scan Engine. X-Allow-Out contains an addition of X-Attribute and the Opt-Body-Type is Attribute-List.

A sample OPTIONS response on SYMCScanResp-AV-DDR is as follows:

```
C: OPTIONS icap://172.16.11.19:1344/SYMCScanResp-AV-DDR ICAP/1.0
C: Host: 172.16.11.19:1344
C: Connection: close
C: Encapsulated: null-body=0
C:
S: ICAP/1.0 200 OK
S: Date: Mon Jun 27 11:50:39 2005 GMT
S: Methods: RESPMOD, FILEMOD
S: Service: Symantec Scan Engine/5.0.0.23
S: Service-ID: SYMCSCANRESP-AV-DDR
S: ISTag: "24E30F27BFBA6209C895B74EF42F6EB6"
S: X-Definition-Info: 20050622.017
S: Max-Connections: 128
S: X-Allow-Out: X-Outer-Container-Is-Mime, X-Infection-Found
S: X-Allow-Out: X-Definition-Info, X-AV-License
S: X-Allow-Out: X-Violations-Found
S: Allow: 204
S: Options-TTL: 3600
S: Preview: 4
S: Transfer-Preview: *
S: X-AV-License: 1
S: Encapsulated: opt-body=0
S: X-Allow-Out: X-Attribute
S: Opt-Body-Type: Attribute-List
S:
S: 211
S: X-ICAP-Attribute-SYMCSCANRESP-AV-DDR
S: Adult Humor
S: Alcohol-Tobacco
S: Anonymous Proxies
S: Crime
S: Drugs/Advocacy
S: Drugs/Non-medical
S: Entertainment/Games
S: Entertainment/Sports
S: Finance
S: Gambling
S: Humor
S: Interactive/Chat
S: Interactive/Mail
S: Intolerance
S: Job Search
S: News
```

```
S: Occult/New Age
S: Prescription Medicine
S: Real Estate
S: Religion
S: Sex/Acts
S: Sex/Attire
S: Sex/Nudity
S: Sex/Personals
S: Sex Education/Advanced
S: Sex Education/Basic
S: Sex Education/Sexuality
S: Travel
S: Vehicles
S: Violence
S: Weapons
S: AllowURLsCategory
S: AllowURLsWithDDRCategory
S: ;
S:
S: 0
S:
```

This response informs the client of the following:

- Besides the OPTIONS method, the only supported method is RESPMOD and FILEMOD.
- The optional 204 shortcut is supported.
- Four bytes of preview information are preferred.
- All files should be sent for preview.
- The data in this response can be cached up to one hour.
- The server supports a maximum of 128 simultaneous connections. The maximum number of simultaneous connections that Symantec Scan Engine supports varies on the operating environment.

You will see new information that starts with Encapsulated: opt-body=0. When scanning for content, an opt-body header is returned that indicates a list of categories that are supported by Symantec Scan Engine. X-Allow-Out contains an addition of X-Attribute and the Opt-Body-Type is Attribute-List.

Scanning HTTP requests (REQMOD)

The legacy service, AVSCANREQ, supports antivirus scanning only. The SYMCScanReq* services handle scanning for viruses and content filtering. You can determine whether you want antivirus scanning only, content filtering only, or antivirus and content-filtering scanning. SYMCScanReq-AV scans only for viruses on POST transactions. SYMCScanReq-AV-URL scans for viruses on POST transactions and content filtering on all transactions.

Symantec Scan Engine supports the following format for specifying scan, repair and block services:

```
icap://server.name:port/symcscanreq-av-url
```

where server.name is the name of the server on which Symantec Scan Engine is running. The port number is optional if Symantec Scan Engine is running on port 1344, which is the default ICAP port.

See "About ICAP responses" on page 29.

REQMOD examples

The ICAP client sends an HTTP request to Symantec Scan Engine, which then returns any of the following responses:

- An unmodified version of the original request
- An HTTP response indicating success or forbidden (for example, virus found or content blocked)
- Error condition (for example, bad gateway)

Examples of REQMOD services are as follows:

- REQMOD antivirus-scanning example
- REQMOD content-filtering scanning example
- REQMOD antivirus and content-filtering scanning example

REQMOD antivirus-scanning example

A sample REQMOD request on SYMCScanReq-AV is as follows:

```
C: REQMOD icap://172.16.11.19:1344/SYMCScanReq-AV ICAP/1.0
C: Host: 172.16.11.19:1344
C: Connection: close
C: Encapsulated: req-hdr=0, req-body=129
C:
```

```
C: POST frere.exe HTTP/1.1
C: Host:icheck.symantec.com
C: Accept: text/html, text/plain
C: Accept-Encoding: compress
C: Pragma: no-cache
C:
C: 3fce
C: Sending chunk of size 16334 bytes
C:
C: 0
C:
S: ICAP/1.0 201 Created
S: ISTag: "B3C20CFCACEDA72CF16F6AEC119B2981"
S: Date: Mon Jun 27 12:45:30 2005 GMT
S: Service: Symantec Scan Engine/5.0.0.23
S: Service-ID: SYMCSCANREQ-AV
S: X-Violations-Found: 1
S:        frere.exe
S:        Jeru.1808.Frere Jac
S:        755
S:        0
S: X-Outer-Container-Is-Mime: 0
S: Encapsulated: res-hdr=0, res-body=110
S:
S: HTTP/1.1 403 Forbidden.
S: Connection: close
S: Content-Length: 673
S: Pragma: no-cache
S: Content-Type: text/html
S:
S: 2a1
S: Getting chunk of size 673 bytes
S:
S: 0
S:
```

This response returns a 201 Created status, which indicates that a problem was found. In this example, a virus was found, which is indicated in the X-Violations-Found section.

REQMOD content-filtering scanning example

A sample REQMOD request on SYMCScanReq-URL is as follows:

```
C: REQMOD icap://172.16.11.19:1344/SYMCSCANREQ-URL ICAP/1.0
C: Host: 172.16.11.19:1344
C: Connection: close
C: Encapsulated: req-hdr=0, null-body=113
C:
C: get / HTTP/1.1
C: Host: www.cnn.com
C: Accept: text/html, text/plain
C: Accept-Encoding: compress
C: Pragma: no-cache
C:
S: ICAP/1.0 201 Created
S: ISTag: "D567F12F8306642DDB9EFAD384BB9B6F"
S: Date: Tue Jul 05 15:41:25 2005 GMT
S: Service: Symantec Scan Engine/5.0.0.24
S: Service-ID: SYMCSCANREQ-URL
S: X-Outer-Container-Is-Mime: 0
S: Encapsulated: res-hdr=0, res-body=110
S: X-Attribute: News
S:
S: HTTP/1.1 403 Forbidden.
S: Connection: close
S: Content-Length: 558
S: Pragma: no-cache
S: Content-Type: text/html
S:
S: 22e
S: Getting chunk of size 558 bytes
S:
S: 0
S:
```

This response returns a 201 Created status, which indicates that a problem was found. In this example, a block message response is returned because the threshold was exceeded for a content category that was configured to be denied in the Symantec Scan Engine administrative interface. In this example, the threshold exceeded the value set for the News category.

REQMOD antivirus and content-filtering scanning example

A sample REQMOD request on SYMCScanReq-AV-URL is as follows:

```
C: REQMOD icap://172.16.11.19:1344/SYMCSCANREQ-AV-URL ICAP/1.0
C: Host: 172.16.11.19:1344
```

```
C: Connection: close
C: Encapsulated: req-hdr=0, req-body=114
C:
C: POST / HTTP/1.1
C: Host:icheck.Symantec.com
C: Accept: text/html, text/plain
C: Accept-Encoding: compress
C: Pragma: no-cache
C:
C: 3ef7
C: Sending chunk of size 16119 bytes
C:
C: 0
C:
S: ICAP/1.0 201 Created
S: ISTag: "29FC058C5B98F78BAB6204F0DF8CE7AA"
S: Date: Tue Jul 05 17:25:05 2005 GMT
S: Service: Symantec Scan Engine/5.0.0.24
S: Service-ID: SYMCSCANREQ-AV-URL
S: X-Violations-Found: 1
S:      index.html/frere.exe
S:      Jeru.1808.Frere Jac
S:      755
S:      0
S: X-Outer-Container-Is-Mime: 0
S: Encapsulated: res-hdr=0, res-body=110
S: X-Attribute: Sex/Acts
S:
S: HTTP/1.1 403 Forbidden.
S: Connection: close
S: Content-Length: 558
S: Pragma: no-cache
S: Content-Type: text/html
S:
S: 22e
S: Getting chunk of size 558 bytes
S:
S: 0
S:
```

This response returns a 201 Created status, which indicates that a problem was found. In this example, a virus was found, which is indicated in the X-Violations-Found section. A block message response is returned because the

threshold was exceeded for a content category that was configured to be denied in the Symantec Scan Engine administrative interface. In this example, the threshold exceeded the value set for the Sex/Acts category.

Scanning HTTP responses (RESPMOD)

The legacy services, AVSCAN and AVSCANRESP, support antivirus scanning only. The SYMCScanResp* services handle scanning for viruses and content filtering. You can determine whether you want antivirus scanning only, content filtering only, or antivirus and content-filtering scanning. SYMCScanResp-AV scans only for viruses. SYMCScanResp-AV-DDR scans for viruses and content filtering.

Symantec Scan Engine supports the following format for specifying scan, repair and block services:

```
icap://server.name:port/symcscanresp-av-ddr
```

where server.name is the name of the server on which Symantec Scan Engine is running. The port number is optional if Symantec Scan Engine is running on port 1344, which is the default ICAP port.

See "About ICAP responses" on page 29.

RESPMOD examples

The ICAP client sends an HTTP response (including the HTTP request headers) to Symantec Scan Engine, which then returns any of the following responses:

- An unmodified version of the original response
- A modified response, indicating what was found
- Error condition (for example, bad gateway)

Examples of RESPMOD services are as follows:

- RESPMOD antivirus-scanning example
- RESPMOD content-filtering scanning example
- RESPMOD antivirus and content-filtering scanning example

RESPMOD antivirus-scanning example

A sample RESPMOD on SYMCScanResp-AV response is as follows:

```
C: RESPMOD icap://172.16.11.19:1344/SYMCSCANRESP-AV ICAP/1.0
C: Host: 172.16.11.19:1344
C: Connection: close
```

```
C: Encapsulated: req-hdr=0, res-hdr=44, res-body=63
C:
C: get / HTTP/1.1
C: Host:icheck.symantec.com
C:
C: HTTP/1.1 200 OK
C:
C: 3ef7
C: Sending chunk of size 16119 bytes
C:
C: 0
C:
S: ICAP/1.0 201 Created
S: ISTag: "1C50CC9255ABDE1C3B0202A416323FE9"
S: Date: Tue Jul 05 15:36:31 2005 GMT
S: Service: Symantec Scan Engine/5.0.0.24
S: Service-ID: SYMCSCANRESP-AV
S: X-Violations-Found: 1
S:      index.html/frere.exe
S:      Jeru.1808.Frere Jac
S:      755
S:      2
S: X-Outer-Container-Is-Mime: 0
S: Encapsulated: res-hdr=0, res-body=57
S:
S: HTTP/1.1 200 OK
S: Via: 1.1 Symantec Scan Engine (ICAP)
S:
S: 1b0
S: Getting chunk of size 432 bytes
S:
S: 0
S:
```

This response returns a 201 Created status, which indicates that a problem was found. In this example, a virus was found, which is indicated in the X-Violations-Found section.

RESPMOD content-filtering scanning example

A sample RESPMOD on SYMCScanResp-DDR response is as follows:

```
C: RESPMOD icap://172.16.11.19:1344/SYMCSCANRESP-DDR ICAP/1.0
C: Host: 172.16.11.19:1344
```

```
C: Connection: close
C: Encapsulated: req-hdr=0, res-hdr=44, res-body=63
C:
C: get / HTTP/1.1
C: Host:icheck.symantec.com
C:
C: HTTP/1.1 200 OK
C:
C: 3ef7
C: Sending chunk of size 16119 bytes
C:
C: 0
C:
S: ICAP/1.0 201 Created
S: ISTag: "7E823734E7070EECD74143AF7CCA7447"
S: Date: Tue Jul 05 15:38:56 2005 GMT
S: Service: Symantec Scan Engine/5.0.0.24
S: Service-ID: SYMCSCANRESP-DDR
S: X-Outer-Container-Is-Mime: 0
S: Encapsulated: res-hdr=0, res-body=83
S: X-Attribute: Sex/Acts
S:
S: HTTP/1.1 200 OK
S: Content-Length: 559
S: Pragma: no-cache
S: Content-Type: text/html
S:
S: 22f
S: Getting chunk of size 559 bytes
S:
S: 0
S:
```

This response returns a 201 Created status, which indicates that a problem was found. In this example, a block message response is returned because the threshold was exceeded for a content category that was configured to be denied in the Symantec Scan Engine administrative interface. This example returns a block message response because the threshold exceeded the value that was set for the Sex/Acts category.

RESPMOD antivirus and content-filtering scanning example

A sample RESPMOD on SYMCScanResp-AV-DDR response is as follows:

```
C: RESPMOD icap://172.16.11.19:1344/SYMCSCANRESP-AV-DDR ICAP/1.0
C: Host: 172.16.11.19:1344
C: Connection: close
C: Encapsulated: req-hdr=0, res-hdr=44, res-body=63
C:
C: get / HTTP/1.1
C: Host:icheck.symantec.com
C:
C: HTTP/1.1 200 OK
C:
C: 3ef7
C: Sending chunk of size 16119 bytes
C:
C: 0
C:
S: ICAP/1.0 201 Created
S: ISTag: "744CB40BD82EC839CD3558834B4C97A5"
S: Date: Tue Jul 05 16:31:50 2005 GMT
S: Service: Symantec Scan Engine/5.0.0.24
S: Service-ID: SYMCSCANRESP-AV-DDR
S: X-Violations-Found: 1
S:      index.html/frere.exe
S:      Jeru.1808.Frere Jac
S:      755
S:      2
S: X-Outer-Container-Is-Mime: 0
S: Encapsulated: res-hdr=0, res-body=83
S: X-Attribute: Sex/Acts
S:
S: HTTP/1.1 200 OK
S: Content-Length: 559
S: Pragma: no-cache
S: Content-Type: text/html
S:
S: 22f
S: Getting chunk of size 559 bytes
S:
S: 0
S:
```

This response returns a 201 Created status, which indicates that a problem was found. In this example, a virus was found, which is indicated in the X-Violations-Found section. A block message response is returned because the

threshold was exceeded for a content category that was configured to be denied in the Symantec Scan Engine administrative interface. In this example, the threshold exceeded the value that was set for the Sex/Acts category.

Scanning non-HTTP data (RESPMOD and FILEMOD)

Not all Symantec Scan Engine clients process HTTP data. Network attached storage devices handle files. Mail transfer agents handle email messages. The files to be scanned can either be sent to Symantec Scan Engine over the network connection or accessed on the file system of the computer that is running Symantec Scan Engine.

See "Scanning HTTP responses (RESPMOD)" on page 59.

See "Local file scanning (FILEMOD)" on page 64.

About network scanning (RESPMOD)

Network scanning is accomplished by mimicking an HTTP response scan. An ICAP request is created that includes manufactured HTTP requests and response headers. The file to be scanned is included as the HTTP message. The HTTP request headers must include a request line with the file name. Typically, the HTTP GET method is used. The HTTP response headers can be the HTTP 200 OK response line.

See "Scanning HTTP responses (RESPMOD)" on page 59.

See "About ICAP responses" on page 29.

Network scanning example

An example of a RESPMOD request is as follows:

```
C: RESPMOD icap://172.16.11.19:1344/SYMCScanResp-AV-DDR ICAP/1.0
C: Host: 172.16.11.19:1344
C: Connection: close
C: Encapsulated: req-hdr=0, res-hdr=52, res-body=71
C:
C: get frere.exe HTTP/1.1
C: Host:icheck.symantec.com
C:
C: HTTP/1.1 200 OK
C:
C: 3fce
C: Sending chunk of size 16334 bytes
```

```
C:
C: 0
C:
S: ICAP/1.0 201 Created
S: ISTag: "334BBC9BDBF997CA16883036DFD9DB81"
S: Date: Mon Jun 27 12:37:13 2005 GMT
S: Service: Symantec Scan Engine/5.0.0.23
S: Service-ID: SYMCSCANRESP-AV-DDR
S: X-Violations-Found: 1
S:        frere.exe
S:        Jeru.1808.Frere Jac
S:        755
S:        2
S: X-Outer-Container-Is-Mime: 0
S: Encapsulated: res-hdr=0, res-body=83
S:
S: HTTP/1.1 200 OK
S: Content-Length: 673
S: Pragma: no-cache
S: Content-Type: text/html
S:
S: 2a1
S: Getting chunk of size 673 bytes
S:
S: 0
S:
```

This response returns a 201 Created status, which indicates that a problem was found. In this example, a virus was found, which is indicated in the X-Violations-Found section.

Local file scanning (FILEMOD)

In the file modification (FILEMOD) mode, an ICAP client sends an ICAP request that contains a file location (with the full path specified) to an ICAP server.

The ICAP server might then do any of the following:

- Send back an ICAP response that contains only headers and no body
- Return an error

This response will contain information about the modification that was performed on the file in-place.

Local file scanning is accomplished by sending an ICAP request with a FILEMOD method to any of the services that handle the FILEMOD method. A Symantec ICAP extension header X-Filepath contains the full path to the file from the point-of-view of the server. If the server cannot access a file at the specified location, a 502 Bad Gateway response is sent. If no X-Filepath header is received, a 400 Bad Request response is sent. The response will be similar to a RESPMOD request for the same file but will contain no HTTP headers and a null-body.

See "About ICAP responses" on page 29.

FILEMOD examples

Examples of RESPMOD services are as follows:

- FILEMOD antivirus-scanning example
- FILEMOD content-filtering scanning example
- FILEMOD antivirus and content-filtering scanning example

FILEMOD antivirus-scanning example

A sample FILEMOD response on SYMCScanResp-AV is as follows:

```
C: FILEMOD icap://172.16.11.19:1344/SYMCScanResp-AV ICAP/1.0
C: Host: 172.16.11.19:1344
C: X-Filepath: d:\vfiles\test2\frere.exe
C: Connection: close
C: Encapsulated: null-body=0
C:
S: ICAP/1.0 403 Forbidden.
S: ISTag: "CC0209F0E11B8594E4BC3043E014D818"
S: Date: Tue Jun 07 14:20:17 2005 GMT
S: Service: Symantec Scan Engine/5.0.0.18
S: Service-ID: SYMCSCANRESP-AV
S: X-Violations-Found: 1
S:       frere.exe
S:       Jeru.1808.Frere Jac
S:       755
S:       2
S: X-Outer-Container-Is-Mime: 0
S: Encapsulated: null-body=0
S:
```

This response returns a 403 Forbidden status, which indicates that a problem was found. In this example, a virus was found, which is indicated in the X-Violations-Found section.

FILEMOD content-filtering scanning example

A sample FILEMOD response on SYMCScanResp-DDR is as follows:

```
C: FILEMOD icap://172.16.11.19:1344/SYMCSCANRESP-DDR ICAP/1.0
C: Host: 172.16.11.19:1344
C: X-Filepath: c:\test_icap\dirty.zip
C: Connection: close
C: Encapsulated: null-body=0
C:
S: ICAP/1.0 403 Forbidden.
S: ISTag: "501260D7CB5A07CBA1CD3F8D787B9960"
S: Date: Tue Jul 05 17:16:37 2005 GMT
S: Service: Symantec Scan Engine/5.0.0.24
S: Service-ID: SYMCSCANRESP-DDR
S: X-Attribute: Sex/Acts
S: X-Outer-Container-Is-Mime: 0
S: Encapsulated: null-body=0
S:
```

This response returns a 403 Forbidden status, which indicates that a problem was found. In this example, a block message response is returned because the threshold was exceeded for a content category that was configured to be denied in the Symantec Scan Engine administrative interface. In this example, the threshold exceeded the value that was set for the Sex/Acts category.

FILEMOD antivirus and content-filtering scanning example

A sample FILEMOD response on SYMCScanResp-AV-DDR is as follows:

```
C: FILEMOD icap://172.16.11.19:1344/SYMCScanResp-AV-DDR ICAP/1.0
C: Host: 172.16.11.19:1344
C: X-Filepath: d:\vfiles\test2\frere.exe
C: Connection: close
C: Encapsulated: null-body=0
C:
S: ICAP/1.0 403 Forbidden.
S: ISTag: "334BBC9BDBF997CA16883036DFD9DB81"
S: Date: Mon Jun 27 12:33:21 2005 GMT
S: Service: Symantec Scan Engine/5.0.0.23
S: Service-ID: SYMCSCANRESP-AV-DDR
```

```
S: X-Violations-Found: 1
S:        frere.exe
S:        Jeru.1808.Frere Jac
S:        755
S:        2
S: X-Outer-Container-Is-Mime: 0
S: Encapsulated: null-body=0
S: X-Attribute: Sex/Acts
S:
```

This response returns a 403 Forbidden status, which indicates that a problem was found. In this example, a virus was found, which is indicated in the X-Violations-Found section. A block message response is returned because the threshold was exceeded for a content category that was configured to be denied in the Symantec Scan Engine administrative interface. In this example, the threshold exceeded the value that was set for the Sex/Acts category.

Chapter 4

Constructing clients using the antivirus client-side API library

This chapter includes the following topics:

General procedure for scanning

The Symantec Scan Engine client API library provides a set of functions to simplify communication with Symantec Scan Engine using ICAP.

The procedure for scanning data is as follows:

- The client calls ScanClientStartUp() once to initialize the client library and to set up the scheduling.
- Files are scanned using either ScanClientScanFile() or ScanClientStreamStart(), ScanClientStreamSendBytes() (as many times as necessary), and ScanClientStreamFinish().
- Information on infections found (if any) is obtained using ScanResultGetNumProblems() and ScanResultGetProblem().
- Completion of scanning for a particular file is indicated using ScanResultsFree().
- Completion of all scanning is indicated (for example, before program termination) using ScanClientShutDown() to free resources.

Sample code is included in the *Symantec Scan Engine Software Developer's Guide* for convenience. The sample code is also included in the distribution package. The sample program demonstrates how to use Symantec Scan Engine API 192.168.1.2 to scan files.

Compiling and linking

Table 4-1 describes requirements for compiling and linking on Solaris, Red Hat Linux, and Windows 2000 Server/Advanced Server.

Table 4-1 Compiling and linking requirements

Platform	Include	Initialize	Link
Solaris	#include "symcsapi.h"	none	libsocket.a libnsl.a (for example, gcc -lsocket -lnsl)
Red Hat Linux	#include "symcsapi.h"	none	libnsl.a (for example, gcc -lnsl)
Windows 2000 Server/Advanced Server	#include "symcsapi.h"	initialize winsock	winsock (for example, WS2_32.LIB)

Note: The code (in all versions of all supported platforms) is compiled with position-independent code so that it can be used in shared libraries.

Compiling on Windows 2000 Server/Advanced Server

To compile on Windows 2000 Server/Advanced Server, use Microsoft Visual Studio 6 or 7.

The source code should include the symcsapi.h file and link to SYMCSAPI.lib. The application should be compiled with multi-threaded runtime libraries. For example, in Microsoft Visual Studio, under Project Settings, click the C/C++ tab, click Code Generation, and click Multithreaded Libraries.

Note: You must link to the winsock library (WS2_32.LIB) and initialize winsock in the source code before scanning files. You must release winsock when all network access is complete (usually just before exiting the program).

Initializing winsock

The client application must initialize winsock before scanning files. The following example demonstrates winsock initialization:

```
#include <windows.h>
.
.
.
// start up winsock
WORD wVersionRequested;
WSADATA wsaData;
  int err;

// Load WinSock, request version 1.0.
wVersionRequested = MAKEWORD(1, 0);
err = WSAStartup(wVersionRequested, &wsaData);
if (err != 0)
{
// ERROR
  }

// Confirm WinSock supports the version we requested.
if (LOBYTE(wsaData.wVersion) != 1 ||
HIBYTE(wsaData.wVersion) != 0)
{
WSACleanup();
// ERROR
  }
```

Shutting down winsock

You must release winsock when all network access is complete.

```
if (WSACleanup() == SOCKET_ERROR)
     ;// ERROR
```

Solaris

If you are using Solaris 9, use gcc compiler version 2.95 or 3.2. If you are using Solaris 10, use gcc compiler version 3.4.6.

The source code that makes calls to the library should include the symcsapi.h header file. In the makefile (or command line) that compiles the program,

libsymcsapi.a should be added and, if it is not already used, -lnsl -lsocket should be added. For example:

gcc mycode.c libsymcsapi.a -lsocket -lnsl

Note: If you are compiling your own multithreaded application, you must define the reentrant symbol (-D_REENTRANT) so that multithreading functions properly. The API libraries are already compiled in this manner.

Red Hat Linux

To compile on Red Hat Linux 7.2 and later, use gcc compiler version 2.95, 3.2, or 3.4.6.

The source code that makes calls to the library should include the symcsapi.h header file. In the makefile (or command line) that compiles the program, libsymcsapi.a should be added and, if it is not already used, -lnsl should be added. For example:

gcc mycode.c libsymcsapi.a -lnsl

Note: If you are compiling your own multithreaded application, you must define the reentrant symbol (-D_REENTRANT) so that multithreading functions properly. The API libraries are already compiled in this manner.

Exceptions and error handling

The Symantec Scan Engine API C library does not throw exceptions or include exception handling. Detected errors are returned as result codes from the functions.

API functions

The API functions are as follows:

- ScanClientStartUp
- ScanClientScanFile
- ScanResultGetNumProblems
- ScanResultGetProblem
- SC_DECODE_DISPOSITION
- ScanResultsFree

- ScanClientShutDown
- ScanClientStreamStart
- ScanClientStreamSendBytes
- ScanClientStreamFinish
- ScanClientStreamAbort
- ScanGetNumConnectErrors
- ScanGetConnectError

File-based scanning

If you are developing a file-based-scanning client, you should use the following functions:

- ScanClientStartUp()
- ScanClientScanFile()
- ScanResultGetNumProblems()
- ScanResultGetProblem()
- ScanResultsFree()
- ScanClientShutDown()

Stream-based scanning

If you are developing a stream-based-scanning client, you should use the following functions:

- ScanClientStartUp()
- ScanClientStreamStart()
- ScanClientStreamSendBytes()
- ScanClientStreamFinish()
- ScanClientStreamAbort()
- ScanResultGetNumProblems()
- ScanResultGetProblem()
- ScanResultsFree()
- ScanClientShutDown()

ScanClientStartUp

The ScanClientStartUp function lets a new client begin submitting files to Symantec Scan Engine.

```
SC_ERROR ScanClientStartUp
(
HSCANCLIENT*client,
LPSTRpszClientConfiguration
)
```

ScanClientStartUp parameters

Table 4-2 lists the parameters that are used.

Table 4-2 ScanClientStartUp parameters

Parameter	Description
client	Address of an HSCANCLIENT variable. The HSCANCLIENT variable is a handle to a status data structure used throughout the scanning process. This variable contains information about the servers that are being used and the scheduling mechanism. Memory is allocated for this data structure by this call and must be freed using ScanClientShutDown() when scanning is completed.

Table 4-2 ScanClientStartUp parameters *(continued)*

Parameter	Description
pszClientConfiguration	A null-terminated string that contains configuration information. Entries are separated with three semicolons (;;;). No spaces are allowed. The following options are supported: ■ Server:ipaddress:port;;;... All Symantec Scan Engines that are used should be listed. At least one server must be listed. ■ FailRetryTime:<seconds> If the client fails to connect to a Symantec Scan Engine, wait <seconds> before trying to connect to that server again (use only the other servers in the meantime, unless they have all failed recently). The default setting is 30 seconds. ■ ReadWriteTime:<seconds> If after <seconds> no response is received from Symantec Scan Engine (when data is being transmitted to Symantec Scan Engine), or if the transmission of data does not complete (when data is being receiving from Symantec Scan Engine), return an error message.

ScanClientStartUp return codes

Table 4-3 lists the return codes (negative values are warnings; positive values are errors).

Table 4-3 ScanClientStartUp return codes

Value	Return code	Description
3	SC_MEMORY_ERROR	A memory allocation error has occurred.
1	SC_INVALID_PARAMETER	A parameter that was passed to the function was invalid.
0	SC_OK	Success.

ScanClientStartUp description

In a multithreaded environment, the client should make a single call to ScanClientStartUp on behalf of all of its threads. This way, all threads use scheduling through a single scheduler rather than through multiple schedulers.

Parameter pszClientConfiguration should look like the following:

Server:1.2.3.4:1344;;;Server:1.2.3.5:1344

At least one Symantec Scan Engine must be specified. Entries are separated with three semicolons. No spaces are allowed. If the default port (1344) is used, the port number (and the colon) can be omitted.

Note: When multiple scan engines are specified, the API provides automatic scheduling across any number of scan engines. The API determines the appropriate scan engine to receive the next file to be scanned, based on the scheduling algorithm. If a scan engine is unreachable or stops responding during a scan, another scan engine is called and the faulty scan engine is taken out of rotation for a period of time (30 seconds is the default setting). If all of the Symantec Scan Engines are out of rotation, the faulty scan engines are called again. The API does not stop trying to contact the Symantec Scan Engines unless five engines do not respond or it appears that a file that is being scanned might have caused more than one engine to stop responding.

ScanClientScanFile

The ScanClientScanFile function is called to have Symantec Scan Engine scan a file for viruses.

Note: The OriginalFileName parameter must be Unicode UTF-8 encoded.

```
SCSCANFILE_RESULT DECLSPEC_SYMCSAPI ScanClientScanFile
(
 HSCANCLIENT   hScanClient,
 LPSTR         pszOriginalFileName,
 LPSTR           pszActualFileName,
 LPSTR           pszOutputFileName,
 LPSTR           pszConfigPolicy,
 LPHSCANRESULTS  phScanResults
)
```

ScanClientScanFile parameters

Table 4-4 lists the parameters that are used.

Table 4-4 ScanClientScanFile parameters

Parameter	Description
hScanClient	An HSCANCLIENT variable that has been initialized. See "ScanClientStartUp" on page 74.
pszOriginalFileName	The name of the file to be scanned. (The original name of the file on the user's computer.) This parameter is ignored if the PassFileByName=1 string is set in pszScanPolicy.
pszActualFileName	The name (path) of the file to scan on the client's computer. This path might be different than pszOriginalFileName. For example, an user may upload a file to be scanned called sample.doc, but the same file may be stored on the ISP computer as temp123.doc. In this case, the pszOriginalFileName is sample.doc, but the pszActualFileName may be /tmp/temp123.doc.
pszOutputFileName	The storage location for the repaired file. (No output file is created unless the input file is infected and the infection is repairable.) This parameter can be any of the following: ■ A null-terminated string that is a path to where the repaired file is to be stored. ■ A char array at least MAX_STRING long with the first byte set to '\0'. The API generates a file name for the repair file. When the function returns, pszOutputFileName is set to the name of the repaired file. **Note:** If this parameter is NULL, the API has Symantec Scan Engine scan the file for viruses but not attempt repair. However, this is not the recommended method for forcing Symantec Scan Engine to scan files for viruses but not attempt repair. You should set the pszScanPolicy to ScanOnly instead. If local scanning options are set using pszScanPolicy, Symantec Scan Engine ignores this parameter and repairs the file in place.

Table 4-4 ScanClientScanFile parameters *(continued)*

Parameter	Description
pszConfigPolicy	A null-terminated string of the form <option:value>;;;<option:value>... No spaces are allowed. The following options are supported: ■ ScanOnly:1: Scan for viruses but do not attempt repair. ■ AlwaysReportDefInfo:1: If a clean file is scanned, a Problem Incident is created with only the virus definitions date and revision number. ■ RepairOnly:1: Attempt to repair infected files but do not delete files that cannot be repaired. ■ ICAPClientIP:<IP address>: Specify IP source address of the encapsulated HTTP request. The IP address must be a dotted-decimal IPv4 address. It is used in ICAP request header X-Client-IP. ■ PassFileByName:1: The file to be scanned is on the same computer as Symantec Scan Engine or can be accessed by network file system (NFS) or another network file protocol. In this case, the file is not sent over a socket. Instead, Symantec Scan Engine opens the file directly for scanning. If a repair is required, the repair is made in the local directory. If this string is set, Symantec Scan Engine uses the pszActualFileName parameter to read the file and uses the ExtensionList= and ExclusionList= options to determine how to handle scanning for a specific file type. This functionality is not available for this release. See "How to maximizing performance" on page 15.
phScanResults	When the function returns, this handle points at a structure that contains information about any problems (infections) found during the scan. If none were found, this parameter is NULL unless the AlwaysReportDefInfo:1 policy is used in the scan. Information about problems is extracted using the described functions. If this parameter is not NULL, the memory must be released using ScanResultsFree().

ScanClientScanFile return codes

Table 4-5 lists the return codes for ScanClientScanFile (negative values indicate that a virus was detected; positive values are errors; zero indicates a clean file).

Table 4-5 ScanClientScanFile return codes

Value	Return code	Description
-3	SCSCANFILE_INF_NO_REP	The file was infected with a virus, but no repair was possible.
-1	SCSCANFILE_INF_REPAIRED	The file was infected, and repair was successful.
0	SCSCANFILE_CLEAN	No virus was found.
1	SCSCANFILE_FAIL_CONNECT	An attempt to connect to a Symantec Scan Engine failed.
2	SCSCANFILE_FAIL_INPUTFILE	A problem was encountered reading the file to be scanned.
3	SCSCANFILE_FAIL_ABORTED	The scan was aborted abnormally.
4	SCSCANFILE_INVALID_PARAM	A function was called with an invalid parameter.
5	SCSCANFILE_FAIL_RECV_FILE	An error occurred when attempting to receive the repaired file.
6	SCSCANFILE_FAIL_MEMORY	A memory allocation error occurred.
7	SCSCANFILE_FAIL_FILE_ACCESS	The server could not access the file to be scanned. This error usually occurs for LOCAL scans when the file permissions are wrong or when the file is not in the path that is specified in the LocalFileScanDir parameter on the server. This error can also occur when the API encounters a problem while writing repaired file data that was received from Symantec Scan Engine to the output file.
10	SCSCANFILE_ERROR_SCANNINGFILE	An internal server error occurred while Symantec Scan Engine was attempting to repair the file.

Table 4-5 ScanClientScanFile return codes *(continued)*

Value	Return code	Description
15	SCSCANFILE_ABORT_NO_AV_ SCANNING_LICENSE	No valid license for antivirus scanning functionality is installed.
17	SCSCANFILE_FAIL_SERVER_TOO_BUSY	The Symantec Scan Engine server is busy and unable to process the scan request.

ScanClientScanFile description

The ScanClientScanFile function determines the appropriate Symantec Scan Engine (when multiple scan engines are running) based on the scheduling algorithm. If a scan engine is unreachable or goes down during a scan, another server is called and the faulty server is taken out of rotation for a period of time. If all scan engines are out of rotation, the faulty servers are called again. The ScanClientScanFile function does not stop trying to contact a scan engine unless five servers are not functioning or it appears that a file that is being scanned might have caused two servers to go down.

ScanResultGetNumProblems

The ScanResultGetNumProblems function indicates the number of infections that are contained in an HSCANRESULT after scanning a file. Use the ScanResultGetProblem() function to get information about the infections that are reported in an HSCANRESULT after scanning a file.

```
SC_ERROR ScanResultGetNumProblems
(
HSCANRESULT hScanResult,
int *nNumProblems
)
```

ScanResultGetNumProblems parameters

Table 4-6 lists the parameters that are used.

Table 4-6 ScanResultGetNumProblems parameters

Parameter	Description
hScanResult	An HSCANRESULT variable that is returned by ScanClientScanFile() or ScanClientStreamFinish().

Table 4-6 ScanResultGetNumProblems parameters *(continued)*

Parameter	Description
nNumProblems	When the function returns, this parameter is set to the number of infections in the scanned file. If the AlwaysReportDefInfo:1 scan policy option is used, at least one (blank) incident is reported, along with the virus definitions date and revision number.

ScanResultGetNumProblems return codes

Table 4-7 lists the return codes (negative values are warnings; positive values are errors).

Table 4-7 ScanResultGetNumProblems return codes

Value	Return code	Description
0	SC_OK	Success.
-1	SC_NULL_PARAMETER	A parameter that was passed to the function is NULL when it should not be.

ScanResultGetProblem

The ScanResultGetProblem function is used to get specific information about a virus.

```
SC_ERROR ScanResultGetProblem
(
HSCANRESULT hScanResult,
int nProblemNum,
int nAttribute,
LPSTR pszValueOut,
LPINT pnValueLengthInOut
)
```

ScanResultGetProblem parameters

Table 4-8 lists the parameters that are used.

Table 4-8 ScanResultGetProblem parameters

Parameter	Description
hScanResult	An HSCANRESULT variable that is returned by ScanClientScanFile() or ScanClientStreamFinish().
nProblemNum	Integer specifying which problem entry is being investigated.
nAttribute	Identifies which attribute is being queried. Valid attributes are as follows: ■ SC_PROBLEM_FILENAME The file name in which the infection occurred. ■ SC_PROBLEM_VIRUSNAME The name of the virus that has infected the file. ■ SC_PROBLEM_NONVIRAL_THREAT_CATEGORY The name of the non-viral threat category. This attribute is sent only if you configure Symantec Scan Engine to send it. For more information, see the *Symantec Scan Engine Implementation Guide*. ■ SC_PROBLEM_VIRUSID The unique numerical ID of the virus. ■ SC_PROBLEM_DISPOSITION Problem resolution. This value is a number (in a string format) that indicates whether the virus was cleaned from the file. ■ SC_PROBLEM_DEFINITION_DATE The date stamp on the virus definitions. ■ SC_PROBLEM_DEFINITION_REV Revision number of the definitions.
pszValueOut	Buffer that holds the attribute value being retrieved.
pnValueLengthInOut	Pointer to the variable that holds the size of the pszValueOut buffer. When the function returns, the size of the attribute string is placed in this variable. You can determine if the buffer was large enough to hold the entire string by seeing whether pnValueLengthInOut is smaller after the call than before. (If the value is smaller, the buffer was large enough.)

ScanResultGetProblem return codes

Table 4-9 lists the ScanResultGet Problem return codes (negative values are warnings; positive values are errors).

Table 4-9 ScanResultGetProblem return codes

Value	Return code	Description
5	SC_OUTOFRANGE_PARAMETER	At least one parameter that was passed to the function was out of range.
1	SC_INVALID_PARAMETER	A parameter that was passed to the function was invalid.
0	SC_OK	Success.

ScanResultGetProblem description

A buffer is supplied to hold the information about the virus, and a pointer is supplied to an integer that holds the size of the buffer. When the function returns, the integer holds the amount of data that was placed in the buffer. If the buffer is not large enough to hold the information, as much information as possible is copied into the buffer. If the value of the integer is the same after the call as it was before the call, the buffer most likely was not large enough to hold the information. See the description of the nAttribute parameter for the type of information that can be retrieved.

SC_DECODE_DISPOSITION

The problem disposition is retrieved with ScanResultGetProblem() in the form of a string. The SC_DECODE_DISPOSITION function is a macro that converts the string to an integer and defines the result codes as integers for easier processing.

```
int SC_DECODE_DISPOSITION( char *pszDisposition )
```

SC_DECODE_DISPOSITION parameters

Table 4-10 lists the parameters that are used.

Table 4-10 SC_DECODE_DISPOSITION parameters

Parameter	Description
pszDisposition	The pszValueOut that is returned from ScanResultGetProblem with nAttribute equal to SC_PROBLEM_DISPOSITION.

SC_DECODE_DISPOSITION return codes

Table 4-11 lists the SC_DECODE_DISPOSITION return codes.

Table 4-11 SC_DECODE_DISPOSITION return codes

Return code	Description
SC_DISP_REPAIRED	The infected file was repaired.
SC_DISP_UNREPAIRED	The infected file was not repaired.
SC_DISP_DELETED	The infected file was deleted.

ScanResultsFree

The ScanResultsFree function is used to free the HSCANRESULT structure. This function must be called when the result structure is no longer needed to free the allocated memory.

```
SC_ERROR ScanResultsFree( HSCANRESULT hScanResult )
```

ScanResultsFree return codes

Table 4-12 lists the ScanResultsFree return codes (negative values are warnings; positive values are errors).

Table 4-12 ScanResultsFree return codes

Value	Return code	Description
0	SC_OK	Success.
-1	SC_NULL_PARAMETER	A parameter that was passed to the function is NULL when it should not be.

ScanClientShutDown

The ScanClientShutDown function is called to clean up after all scanning is complete (for example, before program termination).

```
SC_ERROR ScanClientShutDown( HSCANCLIENT hScanClient )
```

ScanClientShutDown return codes

Table 4-13 lists the ScanClientShutDown return codes (negative values are warnings; positive values are errors).

Table 4-13 ScanClientShutDown return codes

Value	Return code	Description
3	SC_MEMORY_ERROR	A memory allocation error has occurred.
0	SC_OK	Success.

ScanClientStreamStart

The ScanClientStreamStart function is used for scanning streams. It can also be used when you want to accept an input stream for scanning rather than an entire file at one time. For example, if a file is being received through an HTTP stream as a user uploads a file to a Web site, stream scanning can be used.

Note: The OriginalFileName parameter must be Unicode UTF-8 encoded.

```
SC_ERROR DECLSPEC_SYMCSAPI ScanClientStreamStart
(
 HSCANCLIENT hScanClient,
 LPSTR       pszOriginalFileName,
 LPSTR       pszConfigPolicy,
 HSCANSTREAM *phScanStream
)
```

ScanClientStreamStart parameters

Table 4-14 lists the ScanClientStreamStart parameters that are used.

Table 4-14 ScanClientStreamStart parameters

Parameter	Description
hScanClient	This parameter should be set up using ScanClientStartUp().
pszOriginalFileName	The original name of the file to be scanned as it was named on the user's computer.
pszConfigPolicy	A null-terminated string of the form <option:value>;;;<option:value>... No spaces are allowed. See "ScanClientScanFile" on page 76.
phScanStream	Pointer to an HSCANSTREAM variable.

ScanClientStreamStart return codes

Table 4-15 lists the ScanClientStreamStart return codes (negative values are warnings; positive values are errors).

Table 4-15 ScanClientStreamStart return codes

Value	Return code	Description
6	SC_CONNECT_FAILURE	Attempt to connect to a Symantec Scan Engine failed.
3	SC_MEMORY_ERROR	A memory allocation error has occurred.
2	SC_SOCKET_FAILURE	A socket communication error has occurred.
1	SC_INVALID_PARAMETER	A parameter that was passed to the function was invalid.
0	SC_OK	Success.

Setting up stream scanning

The stream scanning feature lets the stream from the user's computer be sent directly to Symantec Scan Engine, rather than first receiving the entire file and then calling ScanClientScanFile().

To set up stream scanning

1. Call ScanClientStreamStart() to initialize the HSCANSTREAM variable.

   ScanClientStreamStart() must be called for each file to be scanned to initialize the HSCANSTREAM variable. HSCANSTREAM variables can be reused only after the stream has been closed with ScanClientStreamFinish() or ScanClientStreamAbort(). The OriginalFileName parameter must be Unicode UTF-8 encoded.

2. Send data to the server in chunks as it is received using ScanClientStreamSendBytes().

3. When all data is sent, call ScanClientStreamFinish().

   To abort the scan between the Start() and Finish() calls, call ScanClientStreamAbort().

ScanClientStreamSendBytes

The ScanClientStreamSendBytes function is used to send chunks of data after HSCANSTREAM has been initialized with ScanClientStreamStart().

See "ScanClientStreamStart" on page 85.

```
SC_ERROR ScanClientStreamSendBytes
(
HSCANSTREAM hStream,
LPBYTE   lpabyData,
DWORD     dwLength
)
```

ScanClientStreamSendBytes parameters

Table 4-16 lists the parameters that are used.

Table 4-16 ScanClientStreamSendBytes parameters

Parameter	Description
hStream	The HSCANSTREAM variable, which must be initialized by a call to ScanClientStreamStart().
lpabyData	Pointer to a buffer that contains the next chunk of data to be sent.
dwLength	Size, in bytes, of the next chunk of data to be sent.

ScanClientStreamSendBytes return codes

Table 4-17 lists the ScanClientStreamBytes return codes (negative values are warnings; positive values are errors).

Table 4-17 ScanClientStreamSendBytes return codes

Value	Return code	Description
2	SC_SOCKET_FAILURE	A socket communication error has occurred.
1	SC_INVALID_PARAMETER	A parameter that was passed to the function was invalid.
0	SC_OK	Success.

ScanClientStreamFinish

The ScanClientStreamFinish function must be called after an entire file has been sent to Symantec Scan Engine to be scanned using ScanClientStreamSendBytes().

See "ScanClientStreamStart" on page 85.

```
SCSCANFILE_RESULT ScanClientStreamFinish
(
HSCANSTREAMhStream,
LPSTRpszOutputFileName,
LPHSCANRESULTSphScanResults
)
```

ScanClientStreamFinish parameters

Table 4-18 lists the parameters that are used.

Table 4-18 ScanClientStreamFinish parameters

Parameter	Description
hStream	The HSCANSTREAM variable, which must be initialized by a call to ScanClientStreamStart().
pszOutputFileName	The storage location for the repaired file. (No output file is created unless the input file is infected and the infection is repairable.) This parameter can be any of the following: ■ A null-terminated string that is a path to where the repaired file is to be stored. ■ A char array at least MAX_STRING long, with the first byte set to '\0'. The API generates a file name for the repair file. When the function returns, pszOutputFileName is set to the name of the repaired file. **Note:** If this parameter is NULL, the API has Symantec Scan Engine scan the file for viruses but not attempt repair. However, this is not the recommended method for forcing Symantec Scan Engine to scan files for viruses but not attempt repair. You should set the pszScanPolicy to ScanOnly instead.
phScanResults	When the function returns, this handle points at a structure that contains information about any problems (infections) found during the scan. If none were found, this parameter is NULL unless the AlwaysReportDefInfo:1 policy is used in the scan. Information about problems is extracted using the described functions. If this parameter is not NULL, the memory must be released using ScanResultsFree().

ScanClientStreamFinish return codes

Table 4-19 lists the ScanClientStreamFinish return codes (negative values are warnings; positive values are errors).

Table 4-19 ScanClientStreamFinish return codes

Value	Return code	Description
-3	SCSCANFILE_INF_NO_REP	The file was infected with a virus, but no repair was possible.
-1	SCSCANFILE_INF_REPAIRED	The file was infected, and repair was successful.
0	SCSCANFILE_CLEAN	No virus was found.
3	SCSCANFILE_FAIL_ABORTED	The scan was aborted abnormally.
4	SCSCANFILE_INVALID_PARAM	Function was called with an invalid parameter.
5	SCSCANFILE_FAIL_RECV_FILE	An error occurred when attempting to receive the repaired file.
6	SCSCANFILE_FAIL_MEMORY	A memory allocation error has occurred.
7	SCSCANFILE_FAIL_FILE_ACCESS	The server could not access the file to be scanned. This error usually occurs for LOCAL scans when the file permissions are wrong or when the file is not in the path that is specified in the LocalFileScanDir parameter on the server. This error can also occur when the API encounters a problem while writing repaired file data that was received from Symantec Scan Engine to the output file.
10	SCSCANFILE_ERROR_SCANNINGFILE	An internal server error occurred while Symantec Scan Engine was attempting to repair the file.

Table 4-19 ScanClientStreamFinish return codes *(continued)*

Value	Return code	Description
15	SCSCANFILE_ABORT_NO_AV_ SCANNING_LICENSE	No valid license for antivirus scanning functionality is installed.

ScanClientStreamAbort

The ScanClientStreamAbort function is called to abort a scan between calls to ScanClientStreamStart() and ScanClientStreamFinish().

```
SC_ERROR ScanClientStreamAbort
(
HSCANSTREAM hStream
)
```

ScanClientStreamAbort return codes

Table 4-20 lists the return codes (negative values are warnings; positive values are errors).

Table 4-20 ScanClientStreamAbort return codes

Value	Return code	Description
1	SC_INVALID_PARAMETER	A parameter that was passed to the function was invalid.
0	SC_OK	Success.
-1	SC_NULL_PARAMETER	A parameter that was passed to the function is NULL when it should not be.

Appendix A

Using the antivirus API

This appendix includes the following topics:

About the sample code

The sample code that is provided in the *Symantec Scan Engine Software Developer's Guide* is for your convenience. Sample code is also included on the Symantec Scan Engine distribution CD. This sample program demonstrates how to use the Symantec Scan Engine API to scan files.

This example demonstrates the use of both file-based and stream-based scanning. File-based scanning is enabled by default.

Sample code

The usage and syntax is as follows:

example <Scan Engine ip>:<Scan Engine port> <input file> [<input file>...]

On Solaris, you should compile the code using code similar to the following:

```
cc example.cpp libsymcsapi.a -lsocket -lnsl -DUNIX
```

On Windows, link to the winsock library. For example:

```
ws2_32.lib
#if defined( WIN32 )
#pragma warning(push,3)
#endif
#include <stdio.h>
```

```
#include <string>
#include "symcsapi.h"
#if defined( WIN32 )
#include <windows.h>
#endif
#if defined( WIN32 )
#include <windows.h>
#endif
// Function Prototypes
void print_prob_info( HSCANRESULTS hResults, int iWhichProblem );
int scanfile( HSCANCLIENT scanclient, char *orig_name, char
  *actual_name);
int main( int argc, char *argv[])
{
HSCANCLIENT scanclient=NULL;
char pszStartUpString[MAX_STRING];
  int i;
#if defined( WIN32 )
// start up winsock
WORD wVersionRequested;
WSADATA wsaData;
  int err;
// Load WinSock, request version 1.0.
wVersionRequested = MAKEWORD(1, 0);
err = WSAStartup(wVersionRequested, &wsaData);
if (err != 0)
{
return 3;// ERROR
  }
// Confirm WinSock supports the version we requested.
if (LOBYTE(wsaData.wVersion) != 1 ||
HIBYTE(wsaData.wVersion) != 0)
{
WSACleanup();
return 3;// ERROR
  }
#endif // defined( WIN32 )
if( argc < 3 )
{
printf( "Usage: %s <ipaddress>:<port> <input-file> [<input-
file>...]\n", argv[0] );
return 1;
  }
```

```
  sprintf( pszStartUpString, "server:%s", argv[1] );
if( ScanClientStartUp( &scanclient, pszStartUpString ) > 0 )
{
printf( "Error in ScanClientStartUp\n");
return 1;
  }
for( i=2; i<argc; i++ )
{
printf( "============================\n");
printf( "Scanning file: %s\n", argv[i] );
printf( "============================\n");
scanfile( scanclient, argv[i], argv[i] );
  }
  ScanClientShutDown( scanclient );
#if defined( WIN32 )
if (WSACleanup() == SOCKET_ERROR)
{
// ERROR
}
#endif
return 0;
}
/*
** Scans a file
**
** Parameters:
**   scanclient: Scan Engine client connection
**   orig_name: Original name of the file
**   actual_name: Name of the file on this machine
**
** Returns:
**   1 for success
**   0 for failure
*/
int scanfile( HSCANCLIENT scanclient, char *orig_name, char
*actual_name)
{
HSCANRESULTS results=NULL;
char repair_file[MAX_STRING];
  int numproblems=0;
// Set up the buffer to accept the repair filename
  repair_file[0] = 0;
#define SCANWHOLEFILE
```

```
#if defined(SCANWHOLEFILE)
// Perform the scan
SCSCANFILE_RESULT answer = ScanClientScanFile( scanclient,
orig_name,
actual_name,
repair_file,
"",
&results
);
#else // SCANWHOLEFILE
  char sendbuff[8 * 1024];
HSCANSTREAM hScanStream = NULL;
if (SC_OK != ScanClientStreamStart(scanclient, orig_name, "",
&hScanStream))
    return 0;
// Open the file and send it
#if defined (WIN32)
HANDLE fd = CreateFile( actual_name, GENERIC_READ, 0, NULL,
OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, NULL);
if (fd == INVALID_HANDLE_VALUE)
{
ScanClientStreamAbort(hScanStream);
return 0;
  }
DWORD iChunkSize;
if (!ReadFile( fd, (void *) sendbuff, sizeof(sendbuff),
&iChunkSize, NULL))
{
CloseHandle( fd );
ScanClientStreamAbort(hScanStream);
return 0;
  }
if (iChunkSize < 0)
{
CloseHandle( fd );
ScanClientStreamAbort(hScanStream);
return 0;
  }
while ( iChunkSize > 0 )
{
if (SC_OK != ScanClientStreamSendBytes(hScanStream,
(LPBYTE)sendbuff, iChunkSize))
{
```

```
CloseHandle( fd );
// Do not call ScanClientStreamAbort here
return 0;
    }
// Keep reading the file
if (!ReadFile( fd, (void *) sendbuff, sizeof(sendbuff),
&iChunkSize, NULL))
{
CloseHandle( fd );
ScanClientStreamAbort(hScanStream);
return 0;
    }
if (iChunkSize < 0)
{
CloseHandle( fd );
ScanClientStreamAbort(hScanStream):
return 0;
}
}
CloseHandle( fd );
#else // if defined(WIN32)
int fd = open(actual_name, O_RDONLY);
if(fd < 0)
{
ScanClientStreamAbort(hScanStream);
return 0;
  }
  int iChunkSize = read( fd,  sendbuff, sizeof( sendbuff));
if( iChunkSize < 0 )
{
close( fd );
ScanClientStreamAbort(hScanStream);
return 0;
  }
while ( iChunkSize > 0 )
{
if (SC_OK != ScanClientStreamSendBytes(hScanStream,
(LPBYTE)sendbuff, iChunkSize))
{
close( fd );
// Do not call ScanClientStreamAbort here
return 0;
    }
```

```
// Keep reading the file
iChunkSize = read( fd,  sendbuff, sizeof( sendbuff));
if( iChunkSize < 0 )
{
close( fd );
ScanClientStreamAbort(hScanStream);
return 0;
}
  }
close( fd );
#endif // if defined(WIN32)
SCSCANFILE_RESULT answer = ScanClientStreamFinish(hScanStream,
repair_file, &results);
#endif // SCANWHOLEFILE
if( answer > 0 )
{
printf( "**** ERROR! Couldn't scan file\n");
return 0;
}
switch( answer )
{
case SCSCANFILE_INF_NO_REP:
printf( "File is infected and cannot be repaired\n");
break;
case SCSCANFILE_INF_REPAIRED:
printf( "File was infected, and has been repaired\n");
break;
case SCSCANFILE_CLEAN:
printf( "File is clean\n");
break;
default:
printf( "ScanClientScanFile returned an unexpected
value\n");
break;
  }
// The results structure will be non-null if a virus was
detected.
if( results )
{
if( strlen( repair_file ) )
{
printf( "Repaired file saved as: %s\n", repair_file );
}
```

```
else
{
printf( "No repair file generated\n");
    }
if( ScanResultGetNumProblems( results, &numproblems ) > 0 )
{
printf( "Error getting number of problems\n");
return 2;
}
printf( "%s had %d infection(s):\n",actual_name,
numproblems );
}
else
{
numproblems = 0;
  }
for( int i=0; i<numproblems; i++ )
{
print_prob_info( results, i );
  }
// Be sure to free the results when done!
  ScanResultsFree( results );
return 1;
}
/*
** Prints scan related information
**
** Parameters:
**   hResults: structure containing the scan results
**   iWhichProblem: scan file problem ID
*/
void print_prob_info( HSCANRESULTS hResults, int iWhichProblem )
{
char attrib[MAX_STRING];
int attrib_size;
  int iDisposition;
attrib_size = MAX_STRING;
ScanResultGetProblem( hResults,
iWhichProblem,
SC_PROBLEM_FILENAME,
attrib,
&attrib_size );
  printf( "File Name: %s\n", attrib );
```

```
attrib_size = MAX_STRING;
ScanResultGetProblem( hResults,
iWhichProblem,
SC_PROBLEM_VIRUSNAME,
attrib,
&attrib_size );
  printf( "Virus Name: %s\n", attrib );
attrib_size = MAX_STRING;
ScanResultGetProblem( hResults,
iWhichProblem,
SC_PROBLEM_VIRUSID,
attrib,
&attrib_size );
  printf( "Virus ID: %s\n", attrib );
attrib_size = MAX_STRING;
ScanResultGetProblem( hResults,
iWhichProblem,
SC_PROBLEM_DISPOSITION,
attrib,
    &attrib_size );
iDisposition = SC_DECODE_DISPOSITION( attrib );
switch( iDisposition )
{
case SC_DISP_UNREPAIRED:
printf( "This infection could not be repaired\n");
break;
case SC_DISP_REPAIRED:
printf( "This infection was repaired\n");
break;
case SC_DISP_DELETED:
printf( "The file with this infection should be deleted\n");
break;
default:
printf( "Unknown Disposition\n");
break;
  }
attrib_size = MAX_STRING;
ScanResultGetProblem( hResults,
iWhichProblem,
SC_PROBLEM_DEFINITION_DATE,
attrib,
&attrib_size );
  printf( "Virus Definitions dated: %s\n", attrib );
```

```
attrib_size = MAX_STRING;
ScanResultGetProblem( hResults,
iWhichProblem,
SC_PROBLEM_DEFINITION_REV,
attrib,
&attrib_size );
printf( "Virus Definitions Revision: %s\n", attrib );
}
```

Index

L

M

N

O

P

R

S

U

V

W

X