Action List Modify Configuration Mode Commands

Action list modify configuration mode commands allow you to configure ACE action lists. An action list is a named group of actions that you associate with a Layer 7 HTTP class map in a Layer 7 HTTP policy map. You can create an action list to modify an HTTP header or to rewrite an HTTP redirect URL for Secure Sockets Layer (SSL).

To create an action list, use the **action-list type modify http** command. The CLI prompt changes to (config-actlist-modify). Use the **no** form of this command to remove the action list from the configuration.

action-list type modify http *name*

no action-list type modify http *name*

Syntax Description

name	Unique name for the action list. Enter an unquoted text string with a maximum of 64 alphanumeric characters.

Command Modes

Configuration mode

Admin and user contexts

Command History

ACE Module Release	Modification
A2(1.0)	This command was introduced.

ACE Appliance Release	Modification
A3(1.0)	This command was introduced.

Usage Guidelines

This command has no usage guidelines.

Examples

To create an action list, enter:

```
host1/Admin(config)# action-list type modify http HTTP_MODIFY_ACTLIST
host1/Admin(config-actlist-modify)#
```

To remove the action list from the configuration, enter:

```
host1/Admin(config)# no action-list type modify http HTTP_MODIFY_ACTLIST
```

Related Commands

show running-config
show stats

(config-actlist-modify) description

(ACE appliance only) To add a description about the action list, use the **description** command. Use the **no** form of this command to remove the description from the action list.

description *text_string*

no description

Syntax Description

text_string	Description for the action list. Enter an unquoted text string with a maximum of 240 alphanumeric characters.

Command Modes

Action list modify configuration mode

Admin and user contexts

Command History

ACE Appliance Release	Modification
A3(2.3)	This command was introduced.

Usage Guidelines

After you create an action list and associate actions with it, you must associate the action list with a Layer 7 policy map. For details, see the *Server Load-Balancing Guide, Cisco ACE Application Control Engine*.

Examples

To add a description for the action list, enter:

```
host1/Admin(config)# action-list type modify http HTTP_MODIFY_ACTLIST
host1/Admin(config-actlist-modify)# description action - delete request
```

To remove the description from the action list, enter:

```
host1/Admin(config-actlist-modify)# no description
```

Related Commands

show action-list

(config-actlist-modify) header delete

To delete an HTTP header from a client request, a server response, or from both, use the **header delete** command in action list modify configuration mode. Use the **no** form of this command to remove the HTTP header delete action from the action list.

header delete {**request** | **response** | **both**} *header-name*

no header delete {**request** | **response** | **both**} *header-name*

Syntax Description

request	Specifies that the ACE delete the header from HTTP request packets from clients.
response	Specifies that the ACE delete the header from HTTP response packets from servers.
both	Specifies that the ACE delete the header from both HTTP request packets and response packets.
header-name	Identifier of the HTTP header that you want to delete. Enter an unquoted text string with a maximum of 255 alphanumeric characters.

Command Modes

Action list modify configuration mode

Admin and user contexts

Command History

ACE Module Release	Modification
A2(1.0)	This command was introduced.

ACE Appliance Release	Modification
A3(1.0)	This command was introduced.

Usage Guidelines

After you create an action list and associate actions with it, you must associate the action list with a Layer 7 policy map. For details, see the *Server Load-Balancing Guide, Cisco ACE Application Control Engine*.

Examples

To delete the Host header from request packets only, enter:

```
host1/Admin(config)# action-list type modify http HTTP_MODIFY_ACTLIST
host1/Admin(config-actlist-modify)# header delete request Host
```

To remove the header delete action from the action list, enter:

```
host1/Admin(config-actlist-modify)# no header delete request Host
```

Related Commands

(config) action-list type modify http
(config-actlist-modify) header insert

(config-actlist-modify) header rewrite

(config-actlist-modify) header insert

When the ACE uses NAT to translate the source IP address of a client to a VIP address, servers need a way to identify that client for the TCP and IP return traffic. To identify a client whose source IP address has been translated using NAT, you can instruct the ACE to insert a generic header and string value in the client HTTP request.

To insert a header name and value in an HTTP request from a client, a response from a server, or both, use the **header insert** command in action list modify configuration mode. Use the **no** form of this command to remove the HTTP header insert action from the action list.

header insert {**request | response | both**} *header-name* **header-value** *expression*

no header insert {**request | response | both**} *header-name* **header-value** *expression*

Syntax Description

request	Specifies that the ACE insert an HTTP header in HTTP request packets from clients.
response	Specifies that the ACE insert an HTTP header in HTTP response packets from servers.
both	Specifies that the ACE insert an HTTP header in both HTTP request packets and response packets.
header-name	Identifier of an HTTP header. Enter an unquoted text string with a maximum of 255 alphanumeric characters.
header-value *expression*	Specifies the value of the HTTP header that you want to insert in request packets, response packets, or both. Enter an unquoted text string with no spaces and a maximum of 255 alphanumeric characters. You can also use the following dynamic replacement strings: • **%is**—Insert the source IP address in the HTTP header. • **%id**—Insert the destination IP address in the HTTP header. • **%ps**—Insert the source port in the HTTP header. • **%pd**—Insert the destination port in the HTTP header.

Command Modes

Action list modify configuration mode

Admin and user contexts

Command History

ACE Module Release	Modification
A2(1.0)	This command was introduced.

ACE Appliance Release	Modification
A3(1.0)	This command was introduced.

Usage Guidelines

After you create an action list and associate actions with it, you must associate the action list with a Layer 7 policy map. For details, see the *Server Load-Balancing Guide, Cisco ACE Application Control Engine*.

With either TCP server reuse or persistence rebalance enabled, the ACE inserts a header in every client request. For information about TCP server reuse, see the *Server Load-Balancing Guide, Cisco ACE Application Control Engine*.

Examples

To include a header insert action for both request and response packets in an action list, enter:

```
host1/Admin(config)# action-list type modify http HTTP_MODIFY_ACTLIST
host1/Admin(config-actlist-modify)# header insert both Host header-value www.cisco.com
```

To remove the insert action from the action list, enter:

```
host1/Admin(config-actlist-modify)# no header insert both Host header-value www.cisco.com
```

Related Commands

(config) action-list type modify http
(config-actlist-modify) header delete
(config-actlist-modify) header rewrite
(config-actlist-modify) ssl header-insert

(config-actlist-modify) header rewrite

To rewrite an HTTP header value in request packets from a client, response packets from a server, or both, use the **header rewrite** command in action list modify configuration mode. Use the **no** form of this command to remove the HTTP header rewrite action from the action list.

header rewrite {**request** | **response** | **both**} *header-name* **header-value** *expression* **replace** *pattern*

no header rewrite {**request** | **response** | **both**} *header-name* **header-value** *expression* **replace** *pattern*

Syntax Description

request	Specifies that the ACE rewrite an HTTP header string in HTTP request packets from clients.
response	Specifies that the ACE rewrite an HTTP header string in HTTP response packets from servers.
both	Specifies that the ACE rewrite an HTTP header string in both HTTP request packets and response packets.
header-name	Identifier of the HTTP header that you want to rewrite. Enter an unquoted text string with a maximum of 255 alphanumeric characters.

header-value *expression*	Specifies the value of the HTTP header that you want to replace in request packets, response packets, or both. Enter a text string from 1 to 255 alphanumeric characters. The ACE supports the use of regular expressions for matching data strings. Use parenthesized expressions for dynamic replacement using %1 and %2 in the replacement pattern. **Note** When matching data strings, the period (.) and question mark (?) characters do not have a literal meaning in regular expressions. Use brackets ([]) to match these symbols (for example, enter www[.]xyz[.]com instead of www.xyz.com). You can also use a backslash (\) to escape a dot (.) or a question mark (?).
replace *pattern*	Specifies the pattern string that you want to substitute for the header value regular expression. For dynamic replacement of the first and second parenthesized expressions from the header value, use %1 and %2, respectively.

Command Modes

Action list modify configuration mode

Admin and user contexts

Command History

ACE Module Release	Modification
A2(1.0)	This command was introduced.

ACE Appliance Release	Modification
A3(1.0)	This command was introduced.

Usage Guidelines

After you create an action list and associate actions with it, you must associate the action list with a Layer 7 policy map. For details, see the *Server Load-Balancing Guide, Cisco ACE Application Control Engine*.

Examples

To include a header replace action for HTTP request packets in an action list, enter:

```
host1/Admin(config)# action-list type modify http HTTP_MODIFY_ACTLIST
host1/Admin(config-actlist-modify)# header rewrite request Host header-value www.cisco.com
replace ?
```

To remove the replace action from the action list, enter:

```
host1/Admin(config-actlist-modify)# no header rewrite request Host header-value
www.cisco.com replace ?
```

Related Commands

(config) action-list type modify http
(config-actlist-modify) header delete
(config-actlist-modify) header insert

(config-actlist-modify) ssl header-insert

To insert HTTP headers containing SSL session information when the ACE receives an HTTP request during a session, use the **ssl header-insert** command. When a client sends encrypted traffic to the ACE in an SSL termination configuration, the ACE terminates the SSL traffic and then sends clear text to the server, which is unaware of the encrypted traffic flowing between the client and the ACE. Using an action list associated with a Layer 7 HTTP load-balancing policy map, you can instruct the ACE to provide the server with the following SSL session information by inserting HTTP headers into the HTTP requests that it receives over the connection:

- Session Parameters—SSL session parameters that the ACE and client negotiate during the SSL handshake.
- Server Certificate Fields—Information regarding the SSL server certificate that resides on the ACE.
- Client Certificate Fields—Information regarding the SSL client certificate that the ACE retrieves from the client when you configure the ACE to perform client authentication.

Use the **no** form of this command to remove the HTTP header insert information.

ssl header-insert {**client-cert** *specific_field* | **server-cert** *specific_field* | **session** *specific_field*} [**prefix** *prefix_string* | **rename** *new_field_name*]

no ssl header-insert {**client-cert** *specific_field* | **server-cert** *specific_field* | **session** *specific_field*} [**prefix** *prefix_string* | **rename** *new_field_name*]

Syntax Description

client-cert *specific_field*	Specifies a client certificate (ClientCert) field name to insert into the HTTP header. See Table 2-6 for a list of the valid client certificate field names.
server-cert *specific_field*	Specifies a server certificate (ServerCert) field name to insert into the HTTP header. See Table 2-7 for a list of the valid server certificate field names.
session *specific_field*	Specifies a session field name to insert into the HTTP header. See Table 2-8 for a list of the valid session field names.
prefix *prefix_string*	(Optional) Inserts a prefix string before the specified field name. For example, if you specify the prefix Acme-SSL for the Authority-Key-Id server certificate field, then the ACE adds the field name as Acme-SSL-ServerCert-Authority-Key-Id. Enter a quoted text string. The maximum combined number of prefix string and field name characters that the ACE permits is 32.
rename *new_field_name*	(Optional) Assigns a new name to the specified field name. Enter an unquoted text string with no spaces. The maximum combined number of field name and prefix string characters that the ACE permits is 32.

Table 2-6 lists the supported SSL client certificate fields. Depending on how the certificate was generated and what key algorithm was used, all of these fields may not be present for the certificate.

Table 2-6 SSL Session Information: SSL Client Certificate Fields

ClientCert Field	Description
Authority-Key-Identifier	X.509 authority key identifier. Format: ASCII string of hexadecimal bytes separated by colons for the X.509 version 3 Authority Key Identifier. Example: ClientCert-Authority-Key-Identifier: 16:13:15:97:FD:8E:16:B9:D2:99
Basic-Constraints	X.509 basic constraints. Format: String that indicates if the certificate subject can act as a certificate authority. Possible values are CA=TRUE or CA=FALSE basic constraints. Example: ClientCert-Basic-Constraints: CA=TRUE
Certificate-Version	X.509 certificate version. Format: Numerical X.509 version (3, 2, or 1), followed by the ASN.1 defined value for X.509 version (2, 1, or 0) in parentheses. Example: ClientCert-Certificate-Version: 3 (0x2)
Data-Signature-Algorithm	X.509 hashing and encryption method. Format: md5WithRSAEncryption, sha1WithRSAEncryption, or dsaWithSHA1 algorithm used to sign the certificate and algorithm parameters. Example: ClientCert-Signature-Algorithm: md5WithRSAEncryption
Fingerprint	SHA1 hash of the certificate. Format: ASCII string of hexadecimal bytes separated by colons. Example: ClientCert-Fingerprint: 64:75:CE:AD:9B:71:AC:25:ED:FE:DB:C7:4B:D4:1:BA
Issuer	X.509 certificate issuer's distinguished name. Format: String of characters representing the certificate authority that issued the certificate. Example: ClientCert-Issuer: CN=Example CA, ST=Virginia, C=US/Email=ca@exampleca.com, 0=Root
Issuer-CN	X.509 certificate issuer's common name. Format: String of characters representing the common name of the certificate issuer. Example: ClientCert-Issuer-CN: www.exampleca.com
Not-After	Date after which the certificate is not valid. Format: Universal time string or generalized time string in the Not After date of the Validity field. Example: ClientCert-Not-After: Dec 12 22:45:13 2014 GMT

Table 2-6 SSL Session Information: SSL Client Certificate Fields (continued)

ClientCert Field	Description
Not-Before	Date before which the certificate is not valid. Format: Universal time string or generalized time string in the Not Before date of the Validity field. Example: ClientCert-Not-Before: Dec 12 22:45:13 2011 GMT
Public-Key-Algorithm	Algorithm used for the public key. Format: rsaEncryption, rsa, or dsaEncryption public key algorithm used to create the public key in the certificate. Example: ClientCert-Public-Key-Algorithm: rsaEncryption
RSA-Modulus	RSA algorithm modulus. Format: RSA algorithm modulus (n) printed in big-endian format hexadecimal, without leading 0x, and lowercase alphanumeric characters separated by a colon (:) character. Together with the exponent (e), this modulus forms the public key portion in the RSA certificate Example: ClientCert-RSA-Modulus: +00:d8:1b:94:de:52:a1:20:51:b1:77
RSA-Exponent	Public RSA exponent. Format: Printed as a whole integer for the RSA algorithm exponent (e). Example: ClientCert-RSA-Exponent: 65537
RSA-Modulus-Size	Size of the RSA public key. Format: Number of bits as a whole integer of the RSA modulus (typically 512, 1024, or 2048) followed by the word bit. Example: ClientCert-RSA-Modulus-Size: 1024 bit
Serial-Number	Certificate serial number. Format: Whole integer value assigned by the certificate authority; this can be any arbitrary integer value. Example: ClientCert-Serial-Number: 2
Signature	Certificate signature. Format: Secure hash of the other fields in the certificate and a digital signature of the hash printed in big-endian format hexadecimal, without leading 0x, and lowercase alphanumeric characters separated by a colon (:) character. Example: ClientCert-Signature: 33:75:8e:a4:05:92:65
Signature-Algorithm	Certificate signature algorithm. Format: md5WithRSAEncryption, sha1WithRSAEncryption, or dsaWithSHA1 for the secure hash algorithm. Example: ClientCert-Signature-Algorithm: md5WithRSAEncryption

Table 2-6 SSL Session Information: SSL Client Certificate Fields (continued)

ClientCert Field	Description
Subject	X.509 subject's distinguished name. Format: String of characters representing the subject that owns the private key being certified. Example: ClientCert-Subject: CN=Example, ST=Virginia, C=US/Email=ca@example.com, 0=Root
Subject-CN	X.509 subject's common name. Format: String of characters that represent the common name of the subject to whom the certificate has been issued. Example: ClientCert-Subject-CN: www.cisco.com
Subject-Key-Identifier	X.509 subject key identifier. Format: ASCII string of hexadecimal bytes separated by colons for the X.509 version 3 subject key identifier. Example: ClientCert-Subject-Key-Identifier: 16:13:15:97:FD:8E:16:B9:D2:99

Table 2-7 lists the supported SSL server certificate fields. Depending on how the certificate was generated and what key algorithm was used, all of these fields may not be present for the certificate.

Table 2-7 SSL Session Information: Server Certificate Fields

ServerCert Field	Description
Authority-Key-Id	X.509 authority key identifier. Format: ASCII string of hex bytes separated by colons for the X.509 version 3 Authority Key Identifier. Example: ServerCert-Authority-Key-Identifier:16:13:15:97:FD:8E:16:B9:D2:99
Basic-Constraints	X.509 basic constraints. Format: String listing whether the certificate subject can act as a certificate authority. Possible values are CA=TRUE or CA=FALSE. Example: ServerCert-Basic-Constraints: CA=TRUE
Certificate-Version	X.509 certificate version. Format: Numerical X.509 version (3, 2, or 1), followed by the ASN.1 defined value for X.509 version (2, 1, or 0) in parentheses. Example: ServerCert-Certificate-Version: 3 (0x2)
Data-Signature-Alg	X.509 hashing and encryption method. Format: md5WithRSAEncryption, sha1WithRSAEncryption, or dsaWithSHA1 algorithm used to sign the certificate and algorithm parameters. Example: ServerCert-Signature-Algorithm: md5WithRSAEncryption

Table 2-7 SSL Session Information: Server Certificate Fields (continued)

ServerCert Field	Description
Fingerprint	SHA1 hash output of the certificate. Format: ASCII string of hexadecimal bytes separated by colons. Example: ServerCert-Fingerprint: 64:75:CE:AD:9B:71:AC:25:ED:FE:DB:C7:4B:D4:1A:BA
Issuer	X.509 certificate issuer's distinguished name. Format: String of characters representing the certificate authority that issued this certificate. Example: ServerCert-Issuer: CN=Example CA, ST=Virginia, C=US/Email=ca@exampleca.com, 0=Root
Issuer-CN	X.509 certificate issuer’s common name. Format: String of characters representing the common name of the certificate issuer. Example: ServerCert-Issuer-CN: www.exampleca.com
Not-After	Date after which the certificate is not valid. Format: Universal time string or generalized time string in the Not After date of the Validity field. Example: ServerCert-Not-After: Dec 12 22:45:13 2014 GMT
Not-Before	Date before which the certificate is not valid. Format: Universal time string or generalized time string in the Not Before date of the Validity field. Example: ServerCert-Not-Before: Dec 12 22:45:13 2011 GMT
Public-Key-Algorithm	Algorithm used for the public key. Format: rsaEncryption, rsa, or dsaEncryption public key algorithm used to create the public key in the certificate. Example: ServerCert-Public-Key-Algorithm: rsaEncryption
RSA-Exponent	Public RSA exponent. Format: Whole integer representing the RSA algorithm exponent (e). Example: ServerCert-RSA-Exponent: 65537
RSA-Modulus	RSA algorithm modulus. Format: RSA algorithm modulus (n) printed in big-endian format hexadecimal, without leading 0x, and lowercase alphanumeric characters separated by a colon (:) character. Together with the exponent (e), this modulus forms the public key portion in the RSA certificate. Example: ServerCert-RSA-Modulus: + 00:d8:1b:94:de:52:a1:20:51:b1:77

Table 2-7 SSL Session Information: Server Certificate Fields (continued)

ServerCert Field	Description
RSA-Modulus-Size	Size of the RSA public key. Format: Number of bits as a whole integer of the RSA modulus (typically 512, 1024, or 2048), followed by the word bit. Example: ServerCert-RSA-Modulus-Size: 1024 bit
Serial-Number	Certificate serial number. Format: Whole integer value assigned by the certificate authority; this can be any arbitrary integer value. Example: ServerCert-Serial-Number: 2
Signature	Certificate signature. Format: Secure hash of the other fields in the certificate and a digital signature of the hash printed in big-endian format hexadecimal, without leading 0x, and lowercase alphanumeric characters and separated by a colon (:) character. Example: ServerCert-Signature: 33:75:8e:a4:05:92:65
Signature-Algorithm	Certificate signature algorithm. Format: md5WithRSAEncryption, sha1WithRSAEncryption, or dsaWithSHA1 for the secure hash algorithm. Example: ServerCert-Signature-Algorithm: nmd5WithRSAEncryption
Subject	X.509 subject's distinguished name. Format: String of characters representing the subject that owns the private key being certified. Example: ServerCert-Subject: CN=Example, ST=Virginia, C=US/Email=ca@example.com, 0=Root
Subject-CN	X.509 subject's common name. Format: String of characters that represents the common name of the certificate issuer. Example: ServerCert-Subject-CN: CN=Example, ST=Virginia, C=US/Email=ca@example.com, 0=Root
Subject-Key-Id	X.509 subject key identifier. Format: ASCII string of hexadecimal bytes separated by colons for the X.509 version 3 subject key identifier. Example: ServerCert-Subject-Key-Identifier: 16:13:15:97: FD:8E:16:B9:D2:99

Table 2-8 lists the supported SSL session fields.

Table 2-8 SSL Session Information: SSL Session Fields

Session Field	Description
Cipher-Key-Size	Symmetric cipher key size. Format: Whole integer that specifies the length in bytes of the public key. Example: Session-Cipher-Key-Size: 32
Cipher-Name	Symmetric cipher suite name. Format: OpenSSL version name of the cipher suite negotiated during the session. Example: Session-Cipher-Name: EXP1024-RC4-SHA
Cipher-Use-Size	Symmetric cipher use size. Format: Whole integer that specifies the length in bytes of the key used for symmetric encryption during this session. Example: Session-Cipher-Use-Size: 7
Id	SSL Session ID. The default is 0. Format: 32-byte session ID negotiated during this session if a session ID is or has been negotiated, printed in big-endian format; hexadecimal without leading 0x and lowercase alphanumeric characters separated by a colon (:). Example: Session-Id: 75:45:62:cf:ee:71:de:ad:be:ef:00:33:ee:23:89:25:75:45:62:cf:ee:71:de:ad:be:ef:00:33:ee:23:89:25
Protocol-Version	Version of SSL or TLS. Format: String that indicates whether SSL or TLS protocol is used followed by a version number. Example: Session-Protocol-Version: TLSv1

Table 2-8 SSL Session Information: SSL Session Fields (continued)

Session Field	Description
Step-Up	Use of SGC or StepUp cryptography. Format: String (yes/no) that indicates whether or not the ACE used Server Gated Cryptography (SGC) or StepUp cryptography to increase the level of security by using 128-bit encryption. Example: Session-Step-Up: YES
Verify-Result	SSL session verify result. Format: String value that indicates the SSL session verify result. Possible values are as follows: • ok—The SSL session is established. • certificate is not yet valid—The client certificate is not yet valid. • certificate is expired—The client certificate has expired. • bad key size—The client certificate has a bad key size. • invalid not before field—The client certificate notBefore field is in an unrecognized format. • invalid not after field—The client certificate notAfter field is in an unrecognized format. • certificate has unknown issuer—The client certificate issuer is unknown. • certificate has bad signature—The client certificate contains a bad signature. • certificate has bad leaf signature—The client certificate contains a bad leaf signature. • unable to decode issuer public key—The ACE is unable to decode the issuer public key. • unsupported certificate—The client certificate is not supported. • certificate revoked— The client certificate has been revoked. • internal error—An internal error exists. Example: Session-Verify-Result: ok

Command Modes

Action list modify configuration mode

Admin and user contexts

Command History

ACE Module/Appliance Release	Modification
A4(1.0)	This command was introduced.

Usage Guidelines

When you instruct the ACE to insert SSL session information, by default the ACE inserts the HTTP header information into the first HTTP request only that it receives over the client connection. When the ACE and client need to renegotiate their connection, the ACE updates the HTTP header information that it send to the server to reflect the new session parameters. You can also instruct the ACE to insert the session information into every HTTP request that it receives over the connection by creating an HTTP parameter map with either the **header modify per-request** or **persistence-rebalance** command enabled. You then reference the parameter map in the policy map that the ACE applies to the traffic. For information about creating an HTTP parameter map, see the *Server Load-Balancing Guide, Cisco ACE Application Control Engine*.

To prevent HTTP header spoofing, the ACE deletes any incoming HTTP headers that match one of the headers that it is going to insert into the HTTP request.

The maximum amount of data that the ACE can insert is 512 bytes. The ACE truncates the data if it exceeds this limit.

Examples

To insert the session Id field with the prefix SSL-, enter:

```
host1/Admin(config)# action-list type modify http HTTP_MODIFY_ACTLIST
host1/Admin(config-actlist-modify)# ssl header-insert session Id prefix SSL-
```

To insert the server certificate Issuer field, enter:

```
host1/Admin(config-actlist-modify)# ssl header-insert server-cert Issuer
```

To insert the client certificate Serial_Number field and rename it Client-Serial-Number, enter:

```
host1/Admin(config-actlist-modify)# ssl header-insert client-cert Serial-Number rename
Client-Serial-Number
```

Related Commands

show stats
(config) action-list type modify http
(config-actlist-modify) header insert

(config-actlist-modify) ssl url rewrite location

To specify the SSL URL, SSL port, and clear port for rewrite, use the **ssl url rewrite location** command. SSL URL rewrite changes the redirect URL from http:// to https:// in the Location response header from the server before sending the response to the client. By doing so, it allows you to avoid nonsecure HTTP redirects because all client connections to the web server will be SSL, thus ensuring the secure delivery of HTTPS content back to the client. Use the **no** form of this command to remove the SSL rewrite specification from the configuration.

ssl url rewrite location *expression* [**clearport** *number*] [**sslport** *number*]

no ssl url rewrite location *expression* [**clearport** *number*] [**sslport** *number*]

Syntax Description

location *expression*	Specifies the rewriting of the URL in the Location response header based on a URL regular expression match. If the URL in the Location header matches the URL regular expression string that you specify, the ACE rewrites the URL from http:// to https:// and rewrites the port number. Enter an unquoted text string with no spaces and a maximum of 255 alphanumeric characters. You can enter a text string with spaces if you enclose the entire string in quotation marks (“). The ACE supports the use of regular expressions for matching data strings. **Note** When matching data strings, the period (.) and question mark (?) characters do not have a literal meaning in regular expressions. Use brackets ([]) to match these symbols (for example, enter www[.]xyz[.]com instead of www.xyz.com). You can also use a backslash (\) to escape a dot (.) or a question mark (?).
clearport *number1*	(Optional) Specifies the clear port number to which the ACE translates the SSL port number before sending a server redirect response to the client. Enter an integer from 1 to 65535. The default is 80.
sslport *number*	(Optional) Specifies the SSL port number from which the ACE translates a clear port number before sending the server redirect response to the client. Enter an integer from 1 to 65535. The default is 443.

Command Modes

Action list modify configuration mode

Admin and user contexts

Command History

ACE Module Release	Modification
A2(1.0)	This command was introduced.

ACE Appliance Release	Modification
A3(1.0)	This command was introduced.

Usage Guidelines

After you create an action list and configure an HTTP redirect URL for SSL, you must associate the action list with a Layer 3 and Layer 4 policy map. For details, see the *Server Load-Balancing Guide, Cisco ACE Application Control Engine*.

Examples

To specify SSL URL rewrite using the default SSL port of 443 and clear port of 80, enter:

```
host1/Admin(config)# action-list type modify http HTTP_MODIFY_ACTLIST
host1/Admin(config-actlist-modify)# ssl url rewrite location www\.website\.com
```

In this case, the ACE rewrites all HTTP redirects to http://www.website.com/ as https://www.website.com/ and forwards them to the client.

Related Commands

(config) action-list type modify http

Action List Optimization Configuration Mode Commands

(ACE appliance only) The action list optimization mode allows you to configure a series of application acceleration and optimization statements. An action list groups a series of individual application acceleration and optimization functions that apply to a specific type of operation. After you enter this command, the system enters the corresponding action list configuration mode.

To access the action list optimization mode, enter the **action-list type optimization http** command. The CLI prompt changes to (config-actlist-optm). To remove an action list optimization selection, use the **no** form of the command. For details about using the commands in the action list optimization mode, see the *Cisco 4700 Series Application Control Engine Appliance Application Acceleration and Optimization Configuration Guide*.

action-list type optimization http *list_name*

no action-list type optimization http *list_name*

Syntax Description

list_name	Name assigned to the action list. Enter a unique name as an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.

Command Modes

Configuration mode

Admin and user contexts

Command History

ACE Appliance Release	Modification
A1(7)	This command was introduced.

Usage Guidelines

The commands in this mode require the loadbalance feature in your user role. For details about role-based access control (RBAC) and user roles, see the *Virtualization Guide, Cisco ACE Application Control Engine*.

The **action-list type** command allows you to configure a series of statements. An action list groups a series of individual functions that apply to a specific type of application acceleration and optimization operation. After you enter this command, the system enters the corresponding action list configuration mode.

After you configure the action list, you associate it with a specific statement in a Layer 7 HTTP optimization policy map. The Layer 7 optimization HTTP policy map activates an optimization HTTP action list that allows you to configure the specified optimization actions.

Examples

To create an optimization HTTP action list, enter:

```
host1/Admin(config)# action-list type optimization http ACT_LIST1
host1/Admin(config-actlist-optm)#
```

To remove the action list from the configuration, enter:

```
host1/Admin(config)# no action-list type optimization http ACT_LIST1
```

Related Commands

show action-list
show running-config
(config) parameter-map type
(config) policy-map

(config-actlist-optm) appscope

(ACE appliance only) To enable AppScope performance monitoring by the optional Cisco AVS 3180A Management Station for use with the ACE, use the **appscope** command. Use the **no** form of this command to disable the AppScope function from the action list.

appscope

no appscope

Syntax Description

This command has no keywords or arguments.

Command Modes

Action list optimization mode

Admin and user contexts

Command History

ACE Appliance Release	Modification
A1(7)	This command was introduced.

Usage Guidelines

The statistical log contains an entry for each ACE optimization request to the server and is used for statistical analysis by the optional Cisco AVS 3180A Management Station. The ACE collects statistical log and sends it to the Cisco AVS 3180A Management Station for loading into the database. For details about the use of the Cisco AVS 3180A Management Station for database, management, and reporting features for the ACE optimization functionality, including AppScope reporting, see the *Cisco 4700 Series Application Control Engine Appliance Application Acceleration and Optimization Configuration Guide*.

To control the AppScope features that measure application acceleration and optimization performance, use the **appscope** commands in parameter map optimization configuration mode. See the "Parameter Map Optimization Configuration Mode Commands" section for details.

To specify the host (the syslog server on the Management Station) that receives the syslog messages sent by the ACE, use the **logging host** configuration command. See the **(config) logging host** command. This command allows you to identify the IP address of the Management Station that will be used as the syslog server. You can specify that the host uses either UDP or TCP to send messages to the syslog server.

Examples

For example, enter:

```
host1/Admin(config)# action-list type optimization http ACT_LIST1
host1/Admin(config-actlist-optm)# appscope
```

To disable the AppScope function from the action list, enter:

```
host1/Admin(config-actlist-optm)# no appscope
```

Related Commands

(config) logging host
(config-parammap-optmz) appscope optimize-rate-percent
(config-parammap-optmz) parameter-summary parameter-value-limit
(config-parammap-optmz) request-grouping-string

(config-actlist-optm) cache

(ACE appliance only) To enable cache optimization for the corresponding URLs, use the **cache** command. Use the **no** form of this command to disable the cache function from the action list.

cache {**dynamic** | **forward** | **forward-with-wait**}

no cache {**dynamic** | **forward** | **forward-with-wait**}

Syntax Description

dynamic	Enables Adaptive Dynamic Caching for the corresponding URLs, even if the expiration settings in the response indicate that the content is dynamic. The expiration of cache objects is controlled by the cache expiration settings based on the time or server load (performance assurance).

forward	Enables the cache forward feature for the corresponding URLs. This keyword allows the ACE to serve the object from its cache (static or dynamic) even when the object has expired if the maximum cache TTL time period has not yet expired (set using the **cache ttl** command in parameter map optimization mode). At the same time, the ACE sends an asynchronous request to the origin server to refresh its cache of the object.
forward-with-wait	Enables the cache forward with wait feature for the corresponding URLs. If the object has expired but the maximum cache TTL time period has not expired (set using the **cache ttl** command in parameter map optimization mode), the ACE sends a request to the origin server for the object. The rest of the users requesting this page will still continue to receive the content from the cache during this time. When the fresh object is returned, it is sent to the requesting user and the cache is also updated. This keyword is similar to the **forward** keyword, except that a single user must wait for the object to be updated before the request is satisfied. This keyword is useful in situations where you are unable to specify the **forward** keyword because the application requires a context for processing and an asynchronous update process is not appropriate.

Command Modes

Action list optimization mode

Admin and user contexts

Command History

ACE Appliance Release	Modification
A1(7)	This command was introduced.

Usage Guidelines

You define the ACE cache object key, cache freshness, and cache request/response policy settings by configuring the cache and cache-policy commands in parameter map optimization configuration mode. See "Parameter Map Optimization Configuration Mode Commands" section for details.

The ACE restricts you from enabling Adaptive Dynamic Caching if you have previously specified either the **delta** command (see "(config-actlist-optm) delta") or the **dynamic etag** command (see "(config-actlist-optm) dynamic etag").

Examples

For example, to enable the cache forward feature for the corresponding URLs, enter:

```
host1/Admin(config)# action-list type optimization http ACT_LIST1
host1/Admin(config-actlist-optm)# cache forward
```

To disable the cache function from the action list, enter:

```
host1/Admin(config-actlist-optm)# no cache forward
```

Related Commands

(config-parammap-optmz) cache key-modifier
(config-parammap-optmz) cache parameter
(config-parammap-optmz) cache ttl
(config-parammap-optmz) cache-policy request
(config-parammap-optmz) cache-policy response

(config-actlist-optm) delta

(ACE appliance only) To enable delta optimization to condense corresponding URLs, use the **delta** command. Use the **no** form of this command to disable delta optimization from the action list.

delta

no delta

Syntax Description

This command has no keywords or arguments.

Command Modes

Action list optimization mode

Admin and user contexts

Command History

ACE Appliance Release	Modification
A1(7)	This command was introduced.

Usage Guidelines

The ACE restricts you from enabling delta optimization if you have previously specified either the **cache dynamic** command (see "(config-actlist-optm) cache") or the **dynamic etag** command (see "(config-actlist-optm) dynamic etag").

Examples

For example, enter:

```
host1/Admin(config)# action-list type optimization http ACT_LIST1
host1/Admin(config-actlist-optm)# delta
```

To disable delta optimization from the action list, enter:

```
host1/Admin(config-actlist-optm)# no delta
```

Related Commands

(config-parammap-optmz) delta

(config-actlist-optm) description

(ACE appliance only) To add a description about the action list, use the **description** command. Use the **no** form of this command to remove the description from the action list.

description *text_string*

no description

Syntax Description

text_string	Description for the action list. Enter an unquoted text string with a maximum of 240 alphanumeric characters.

Command Modes

Action list modify configuration mode

Admin and user contexts

Command History

ACE Appliance Release	Modification
A3(2.3)	This command was introduced.

Usage Guidelines

After you create an action list and associate actions with it, you must associate the action list with a Layer 7 policy map. For details, see the *Server Load-Balancing Guide, Cisco ACE Application Control Engine*.

Examples

To add a description for the action list, enter:

```
host1/Admin(config)# action-list type optimization http ACT_LIST1
host1/Admin(config-actlist-optm)# description action - delta
```

To remove the description from the action list, enter:

```
host1/Admin(config-actlist-optm)# no description
```

Related Commands

show action-list

(config-actlist-optm) dynamic etag

(ACE appliance only) To enable just-in-time object acceleration for the corresponding URLs, use the **dynamic etag** command. Use the **no** form of this command to disable just-in-time object acceleration from the action list.

dynamic etag

no dynamic etag

Syntax Description

This command has no keywords or arguments.

Command Modes

Action list optimization mode

Admin and user contexts

Command History

ACE Appliance Release	Modification
A1(7)	This command was introduced.

Usage Guidelines

The ACE restricts you from enabling just-in-time object acceleration if you have previously specified either the **cache dynamic** command (see "(config-actlist-optm) cache") or the **delta** command (see "(config-actlist-optm) delta").

Examples

For example, enter:

```
host1/Admin(config)# action-list type optimization http ACT_LIST1
host1/Admin(config-actlist-optm)# dynamic etag
```

To disable just-in-time object acceleration from the action list, enter:

```
host1/Admin(config-actlist-optm)# no dynamic etag
```

Related Commands

This command has no related commands.

(config-actlist-optm) flashforward

(ACE Appliance only) To enable FlashForward for the corresponding URLs and to transform embedded objects, use the **flashforward** command. Use the **no** form of this command to disable FlashForward from the action list.

flashforward

no flashforward

Syntax Description

This command has no keywords or arguments.

Command Modes

Action list optimization mode

Admin and user contexts

Command History

ACE Appliance Release	Modification
A1(7)	This command was introduced.

Usage Guidelines

The **flashforward** and **flashforward-object** commands cannot be configured in the same optimization action list; these two commands are mutually exclusive.

Examples

For example, enter:

```
host1/Admin(config)# action-list type optimization http ACT_LIST1
host1/Admin(config-actlist-optm)# flashforward
```

To disable FlashForward from the action list, enter:

```
host1/Admin(config-actlist-optm)# no flashforward
```

Related Commands

(config-actlist-optm) flashforward-object
(config-parammap-optmz) flashforward refresh-policy
(config-parammap-optmz) rebase

(config-actlist-optm) flashforward-object

(ACE appliance only) To enable FlashForward static caching for the corresponding URLs, use the **flashforward-object** command. Use the **no** form of this command to disable FlashForward static caching from the action list.

flashforward-object

no flashforward-object

Syntax Description

This command has no keywords or arguments.

Command Modes

Action list optimization mode

Admin and user contexts

Command History

ACE Appliance Release	Modification
A1(7)	This command was introduced.

Usage Guidelines

The **flashforward-object** and **flashforward** commands cannot be configured in the same optimization action list; these two commands are mutually exclusive.

Examples

For example, enter:

```
host1/Admin(config)# action-list type optimization http ACT_LIST1
host1/Admin(config-actlist-optm)# flashforward-object
```

To disable FlashForward static caching from the action list, enter:

```
host1/Admin(config-actlist-optm)# no flashforward-object
```

Related Commands

(config-actlist-optm) flashforward
(config-parammap-optmz) flashforward refresh-policy
(config-parammap-optmz) rebase