

Parameter Map Connection Configuration Mode Commands

Parameter map connection configuration mode commands allow you to define a connection-type parameter map. After you create the connection parameter map, you can configure TCP, IP, and other settings for the map. To create the connection parameter map and access parameter map connection configuration mode, use the **parameter-map type connection** command in configuration mode. The prompt changes to (config-parammap-conn). Use the **no** form of this command to remove the parameter map from the configuration.

parameter-map type connection *name*

no parameter-map type connection *name*

Syntax Description	<i>name</i>	Name assigned to the parameter map. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.
---------------------------	-------------	---

Command Modes	Configuration mode Admin and user contexts
----------------------	---

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines The commands in this mode require the connection feature in your user role. For details about role-based access control (RBAC) and user roles, see the *Cisco 4700 Series Application Control Engine Appliance Virtualization Configuration Guide*.

After you create and configure a parameter map, you must associate the parameter map with a policy map to activate it. For details, see the [\(config-pmap-c\) connection advanced-options](#) command in the “Policy Map Configuration Mode Commands” section.

Examples To create a connection parameter map called TCP_MAP, enter:

```
host1/Admin(config)# parameter-map type connection TCP_MAP
host1/Admin(config-parammap-conn)#
```

To delete the connection parameter map, enter:

```
host1/Admin(config)# no parameter-map type connection TCP_MAP
```

Related Commands [\(config\) parameter-map type](#)
[\(config-pmap-c\) connection advanced-options](#)
[show parameter-map](#)

(config-parammap-conn) exceed-mss

To configure the ACE to allow segments that exceed the maximum segment size (MSS), use the **exceed-mss** command. Use the **no** form of this command to reset the ACE to its default of discarding segments that exceed the MSS.

exceed-mss {allow | drop}

no exceed-mss

Syntax Description	allow	drop
	Permits segments that exceed the maximum segment size.	Discards segments that exceed the maximum segment size. This is the default.

Command Modes
Parameter map connection configuration mode
Admin and user contexts

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines
This command has no usage guidelines.

Examples
To configure the ACE to allow segments that exceed the MSS, enter:

```
host1/Admin(config-parammap-conn)# exceed-mss allow
```

To configure the ACE to discard segments that exceed the MSS, enter:

```
host1/Admin(config-parammap-conn)# exceed-mss drop
```

To reset the ACE behavior to the default of discarding segments that exceed the MSS, enter:

```
host1/Admin(config-parammap-conn)# no exceed-mss allow
```

Related Commands
[\(config-parammap-conn\) set tcp mss](#)
[show parameter-map](#)

(config-parammap-conn) nagle

To enable Nagle's algorithm, use the **nagle** command. By default, this command is disabled. Nagle's algorithm instructs a sender to buffer any data to be sent until all outstanding data has been acknowledged or until there is a full segment of data to send. Use the **no** form of this command to disable Nagle's algorithm.

nagle

no nagle

Syntax Description This command has no keywords or arguments.

Command Modes Parameter map connection configuration mode
Admin and user contexts

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines Nagle's algorithm automatically concatenates a number of small buffer messages that are transmitted over the TCP connection. This process increases throughput by decreasing the number of segments that need to be sent over the network. However, the interaction between Nagle's algorithm and the TCP delay acknowledgment may increase latency in your TCP connection. You should disable Nagle's algorithm if you notice delays in your TCP connection.

Examples To enable Nagle's algorithm, enter:

```
host1/Admin(config-parammap-conn) # nagle
```

To disable Nagle's algorithm, enter:

```
host1/Admin(config-parammap-conn) # no nagle
```

Related Commands [show parameter-map](#)

(config-parammap-conn) random-sequence-number

To enable TCP sequence number randomization, use the **random-sequence-number** command. This feature is enabled by default. Use the **no** form of this command to disable sequence number randomization.

random-sequence-number

no random-sequence-number

Syntax Description This command has no keywords or arguments.

Command Modes Parameter map connection configuration mode
Admin and user contexts

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines Randomizing TCP sequence numbers makes it more difficult for a hacker to guess or predict the next sequence number in a TCP connection.

Examples To enable sequence number randomization, enter:

```
host1/Admin(config-parammap-conn) # random-sequence-number
```

To disable sequence number randomization, enter:

```
host1/Admin(config-parammap-conn) # no random-sequence-number
```

Related Commands [show parameter-map](#)

(config-parammap-conn) reserved-bits

To configure how an ACE handles segments with the reserved bits set in the TCP header, use the **reserved-bits** command. Use the **no** form of this command to reset the ACE to its default of clearing reserved bits set in the TCP header of a segment.

```
reserved-bits { allow | clear | drop }
```

```
no reserved-bits
```

Syntax Description	allow	Permits segments with the reserved bits set in the TCP header.
	clear	Clears the reserved bits in the TCP header and allows the segment. This is the default.
	drop	Discards segments with reserved bits set in the TCP header.

Command Modes	Parameter map connection configuration mode Admin and user contexts
---------------	--

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines	The six reserved bits in the TCP header are for future use and have a value of 0.
------------------	---

Examples	To configure the ACE to allow segments with the reserved bits set in the TCP header, enter: <pre>host1/Admin(config-parammap-conn)# reserved-bits allow</pre> <p>To reset the ACE to its default of clearing reserved bits set in the TCP header of a segment, enter: <pre>host1/Admin(config-parammap-conn)# no reserved-bits allow</pre></p>
----------	--

Related Commands	show parameter-map
------------------	------------------------------------

(config-parammap-conn) set ip tos

To set the type of service (ToS) for packets in a particular traffic class, use the **set ip tos** command. Use the **no** form of the command to instruct the ACE to not rewrite the IP ToS value.

set ip tos *number*

no set ip tos

Syntax Description	<i>number</i>	Packet ToS value. Enter an integer from 0 to 255.
---------------------------	---------------	---

Command Modes	Parameter map connection configuration mode Admin and user contexts
----------------------	--

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines	The ToS for a packet determines how the network handles the packet and balances its precedence, delay, throughput, and reliability. This information resides in the IP header. For details about the ToS byte, see RFCs 791, 1122, 1349, and 3168.
-------------------------	---

Examples	To set a packet's ToS value to 20, enter: <pre>host1/Admin(config-parammap)# set ip tos 20</pre> To instruct the ACE to ignore the ToS of a packet, enter: <pre>host1/Admin(config-parammap)# no set ip tos</pre>
-----------------	--

Related Commands	show parameter-map
-------------------------	------------------------------------

(config-parammap-conn) set tcp ack-delay

To configure an ACK delay, use the **set tcp ack-delay** command. You can configure the ACE to delay sending the ACK from a client to a server. Some applications delay the ACK for best performance. To reset the ACK delay timer to the default value of 200 ms, use the **no** form of the command.

set tcp ack-delay *number*

no set tcp ack-delay

Syntax Description

<i>number</i>	Delay time for sending an ACK from a client to a server. Enter an integer from 0 to 400 ms. The default is 200 ms.
---------------	--

Command Modes

Connection parameter-map configuration mode

Command History

Release	Modification
3.0(2)	This command was introduced.

Usage Guidelines

Delaying the ACK can help reduce congestion by sending one ACK for multiple segments rather than sending an ACK for each segment.

Examples

To delay sending an ACK for 400 ms, enter:

```
host1/Admin(config-parammap-conn)# set tcp ack-delay 400
```

To reset the ACK delay timer to the default of 200 ms, enter:

```
host1/Admin(config-parammap-conn)# no set tcp ack-delay
```

Related Commands

[show parameter-map](#)

(config-parammap-conn) set tcp mss

To set a range of values for the TCP maximum segment size (MSS), use the **set tcp mss** command. Use the **no** form of this command to reset the minimum MSS to the default of 536 bytes and the maximum MSS to the default of 1380.

```
set tcp mss min number1 max number2
```

```
no set tcp mss
```

Syntax Description

min <i>number1</i>	Specifies the smallest segment size in bytes that the ACE will accept. Enter an integer from 0 to 65535. The default is 536 bytes. If the ACE receives a segment smaller than the configured minimum size, the appliance discards the segment.
max <i>number2</i>	Specifies the largest segment size in bytes that the ACE will accept. Enter an integer from 0 to 65535. The default is 1380 bytes. If the ACE receives a segment larger than the configured maximum size, the appliance discards the segment.

Command Modes

Parameter map connection configuration mode
Admin and user contexts

Command History

Release	Modification
A1(7)	This command was introduced.

Usage Guidelines

The MSS is the largest amount of TCP data that the ACE accepts in one segment. To prevent the transmission of many smaller segments or very large segments that may require fragmentation, you can set the minimum and maximum acceptable sizes of the MSS.

Both the host and the server can set the MSS when they first establish a connection. If either maximum value exceeds the value that you set with the **set tcp mss max** command, then the ACE overrides the maximum value and inserts the value that you set. If either maximum value is less than the value that you set with the **set tcp mss min** command, then the ACE overrides the maximum value and inserts the minimum value (the minimum value is actually the smallest maximum allowed). For example, if you set a maximum value of 1200 bytes and a minimum value of 400 bytes, when a host requests a maximum value of 1300 bytes, then the ACE alters the packet to request 1200 bytes (the maximum). If another host requests a maximum value of 300 bytes, then the ACE alters the packet to request 400 bytes (the minimum).

The default of 1380 bytes allows room for header information so that the total packet size does not exceed 1500 bytes, which is the default MTU for Ethernet. See the following calculation:

1380 data + 20 TCP + 20 IP + 24 AH + 24 ESP_CIPHER + 12 ESP_AUTH + 20 IP = 1500 bytes

If the host or server does not request an MSS, the ACE assumes that the RFC 793 default value of 536 bytes is in effect.

If you set the MSS to be greater than 1380, packets might become fragmented, depending on the MTU size (which is 1500 by default for Ethernet). Large numbers of fragments can impact the performance of the ACE. Setting the minimum size prevents the TCP server from sending many small TCP data packets to the client and impacting the performance of the server and the network.

Examples

To set the minimum acceptable MSS value to 768 bytes and the maximum acceptable MSS value to 1500, enter:

```
host1/Admin(config-parammap-conn)# set tcp mss min 768 max 1500
```

To reset the minimum MSS to the default of 536 bytes and the maximum MSS to the default of 1380, enter:

```
host1/Admin(config-parammap-conn)# no set tcp mss
```

Related Commands

[\(config-parammap-conn\) exceed-mss](#)
[show parameter-map](#)

(config-parammap-conn) set tcp syn-retry

To set the maximum number of attempts that the ACE can take to transmit a TCP segment, use the **set tcp syn-retry** *number* command. Use the **no** form of this command to reset the maximum number of TCP SYN retries to the default of 4.

set tcp syn-retry *number*

no set tcp syn-retry

Syntax Description	<i>number</i>	Number of SYN retries. Enter an integer from 1 to 6. The default is 4.
--------------------	---------------	--

Command Modes	Parameter map connection configuration mode Admin and user contexts
---------------	--

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines	This command has no usage guidelines.
------------------	---------------------------------------

Examples	To set the maximum number of attempts that the ACE takes to transmit a TCP segment to 3, enter: <pre>host1/Admin(config-parammap-conn) # set tcp syn-retry 3</pre> To reset the maximum number of TCP SYN retries to the default of 4, enter: <pre>host1/Admin(config-parammap-conn) # no set tcp syn-retry</pre>
----------	--

Related Commands	show parameter-map
------------------	------------------------------------

(config-parammap-conn) set tcp timeout

To configure a timeout for TCP embryonic connections (connections that result from an incomplete three-way handshake) and half-closed connections (connections where the client has sent a FIN and the server has not responded), use the **set tcp timeout** command. Use the **no** form of this command to reset TCP timeout values to their default settings.

```
set tcp timeout {embryonic seconds | half-closed seconds}
```

```
no set tcp timeout {embryonic | half-closed}
```

Syntax Description

embryonic	Specifies the timeout for embryonic connections.
<i>seconds</i>	Time in seconds after which the ACE times out an embryonic connection. Enter an integer from 0 to 4294967295. The default is 5 seconds. A value of 0 specifies that the ACE never time out an embryonic connection.
half-closed	Specifies the timeout for half-closed connections.
<i>seconds</i>	Time in seconds after which the ACE times out a half-closed connection. Enter an integer from 0 to 4294967295. The default is 3600 seconds (1 hour). A value of 0 specifies that the ACE never time out a half-closed TCP connection.

Command Modes

Parameter map connection configuration mode
Admin and user contexts

Command History

Release	Modification
A1(7)	This command was introduced.

Usage Guidelines

This command has no usage guidelines.

Examples

To set the TCP timeout for embryonic connections to 24 seconds, enter:

```
host1/Admin(config-parammap-conn) # set tcp timeout embryonic 24
```

To reset the TCP half-closed connection timeout to the default of 600 seconds, enter:

```
host1/Admin(config-parammap-conn) # no set tcp timeout half-closed
```

Related Commands

[show parameter-map](#)

(config-parammap-conn) set tcp wan-optimization

To control how the ACE applies TCP optimizations to packets on a connection associated with a Layer 7 policy map using a round-trip time (RTT) value, use the **set tcp wan-optimization** command. Use the **no** form of this command to restore the ACE behavior to the default of not optimizing TCP connections.

set tcp wan-optimization rtt *number*

no set tcp wan-optimization rtt *number*

Syntax Description

<i>number</i>	The RTT value. Enter an integer from 0 to 65535. The default is 65535.
---------------	--

Command Modes

Parameter map connection configuration mode
Admin and user contexts

Command History

Release	Modification
3.0(2)	This command was introduced.

Usage Guidelines

This command allows you to control how the ACE applies TCP optimizations to packets on a connection associated with a Layer 7 policy map using the following RTT value:

- For a value of 0, the ACE applies TCP optimizations to packets for the life of a connection
- For a value of 65535 (the default), the ACE performs normal operations (no optimizations) for the life of a connection
- For values from 1 to 65534, the ACE applies TCP optimizations to packets based on the client RTT to the ACE as follows:
 - If the actual client RTT is less than the configured RTT, the ACE performs normal operations for the life of the connection
 - If the actual client RTT is greater than or equal to the configured RTT, the ACE performs TCP optimizations on the packets for the life of a connection

TCP optimizations include the following connection parameter-map configuration mode operations:

- Nagle optimization algorithm
- Slowstart connection behavior
- Acknowledgement (ACK) delay timer
- Window-scale factor
- Retry settings

Examples

To set the RTT to 0 to apply TCP optimizations to packets for the life of a connection, enter:

```
host1/Admin(config-parammap-conn)# set tcp wan-optimization rtt 0
```

To restore the ACE behavior to the default of not optimizing TCP connections, enter:

```
host1/Admin(config-parammap-conn)# no set tcp wan-optimization rtt
```

Related Commands

[show parameter-map](#)

(config-parammap-conn) set tcp window-scale

To configure a TCP window-scale factor for network paths with high-bandwidth, long-delay characteristics, use the **set tcp window-scale** command. Use the **no** form of this command to reset the window-scale factor to its default setting.

set tcp window-scale *number*

no set tcp window-scale

Syntax Description	<i>number</i>	Window-scale factor. Enter an integer from 0 to 14. The default is 0.
---------------------------	---------------	---

Command Modes	Parameter map connection configuration mode Admin and user contexts
----------------------	--

Command History	Release	Modification
	3.0(2)	This command was introduced.

Usage Guidelines	<p>The TCP window scaling feature adds support for the Window Scaling option in RFC 1323. We recommend increasing the window size to improve TCP performance in network paths with large bandwidth, long-delay characteristics. This type of network is called a long fat network (LFN).</p> <p>The window scaling extension expands the definition of the TCP window to 32 bits and then uses a scale factor to carry this 32-bit value in the 16-bit window field of the TCP header. You can increase the window size to a maximum scale factor of 14. Typical applications use a scale factor of 3 when deployed in LFNs.</p>
-------------------------	--

Examples	<p>To set the TCP window-scale factor to 3, enter:</p> <pre>host1/Admin(config-parammap-conn)# set tcp window-scale 3</pre> <p>To reset the TCP window-scale factor to the default of 0, enter:</p> <pre>host1/Admin(config-parammap-conn)# no set tcp window-scale</pre>
-----------------	---

Related Commands	show parameter-map
-------------------------	------------------------------------

(config-parammap-conn) set timeout inactivity

To configure the connection inactivity timer, use the **set timeout inactivity** command. Use the **no** form of this command to reset the timeout inactivity values to the default ICMP, TCP, and UDP settings.

set timeout inactivity *seconds*

no set timeout inactivity

Syntax Description

inactivity	Specifies the timeout for idle TCP connections.
<i>seconds</i>	Time period after which the ACE disconnects idle established connections. Enter an integer from 0 to 4294967294. A value of 0 specifies that the ACE never times out a TCP connection. Default settings are as follows: <ul style="list-style-type: none"> • ICMP—2 seconds • TCP—3600 seconds (1 hour) • UDP—120 seconds (2 minutes)

Command Modes

Parameter map connection configuration mode
Admin and user contexts

Command History

Release	Modification
A1(7)	This command was introduced.

Usage Guidelines

The ACE uses the connection inactivity timer to disconnect established ICMP, TCP, and UDP connections that have remained idle for the duration of the specified timeout period. The ACE rounds up the configured timeout value to the nearest 30-second interval.

Examples

To specify that the ACE disconnect idle established TCP connections after 2400 seconds, enter:

```
host1/Admin(config-parammap-conn) # set timeout inactivity 2400
```

To reset the ICMP, TCP, and UDP inactivity timeout to the default values, enter:

```
host1/Admin(config-parammap-conn) # no set timeout inactivity
```

Related Commands

[show parameter-map](#)

(config-parammap-conn) slowstart

To enable the slow start algorithm, use the **slowstart** command. This feature is enabled by default. Use the **no** form of this command to disable the slow start algorithm.

slowstart

no slowstart

Syntax Description This command has no keywords or arguments.

Command Modes Parameter map connection configuration mode
Admin and user contexts

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines The slow start algorithm is a congestion avoidance method in which TCP increases its window size as ACK handshakes arrive. It operates by observing that the rate at which new segments should be injected into the network is the rate at which the acknowledgments are returned by the host at the other end of the connection. For further details about the TCP slow start algorithm, see RFC 3390.

Examples To enable the slow start algorithm, enter:

```
host1/Admin(config-parammap-conn)# slowstart
```

To disable the slow start algorithm, enter:

```
host1/Admin(config-parammap-conn)# no slowstart
```

Related Commands [show parameter-map](#)

(config-parammap-conn) syn-data

To set the ACE to discard SYN segments with data, use the **syn-data** command. Use the **no** form of this command to reset the ACE to its default of allowing SYN segments that contain data.

syn-data { **allow** | **drop** }

no syn-data

Syntax Description	allow	drop
	Permits the SYN segments that contain data and flags them for data processing. This is the default.	Discards the SYN segments that contain data.

Command Modes	Parameter map connection configuration mode Admin and user contexts
---------------	--

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines	Occasionally, the ACE may receive a SYN segment that contains data. You can configure the ACE to either discard the segment or flag the segment for data processing.
------------------	--

Examples To instruct the ACE to discard segments that contain data, enter:

```
host1/Admin(config-parammap-conn)# syn-data drop
```

To reset the ACE to its default of allowing SYN segments that contain data, enter:

```
host1/Admin(config-parammap-conn)# no syn-data
```

Related Commands	show parameter-map
------------------	------------------------------------

(config-parammap-conn) tcp-options

To specify a range of TCP options not explicitly supported by the ACE, or allow or clear explicitly supported TCP options specified in a SYN segment, use the **tcp-options** command. Use the **no** form of this command to remove a TCP option range from the configuration or reset the ACE to its default of clearing the specific TCP options.

```
tcp-options {range number1 number2 {allow | drop}} | {selective-ack | timestamp |
window-scale {allow | clear}}
```

```
no tcp-options {range number1 number2 {allow | drop}} | {selective-ack | timestamp |
window-scale {allow | clear}}
```

Syntax Description	range <i>number1 number2</i>	Specifies the TCP options not explicitly supported by the ACE using a range of option numbers. The arguments are as follows:
		<ul style="list-style-type: none"> <i>number1</i>—Specifies the lower limit of the TCP option range. Enter either 6 or 7 or an integer from 9 to 255. See the “Usage Guidelines” section for the available TCP options. <i>number2</i>—Specifies the upper limit of the TCP option range. Enter 6 or 7 or an integer from 9 to 255. See the “Usage Guidelines” section for the available TCP options.
	allow	Allows any segment with the specified option set.
	drop	Causes the ACE to discard any segment with the specified option set.
	selective-ack	Allows the ACE to inform the sender about all segments that it received. The sender needs to retransmit the lost segments, rather than wait for a cumulative acknowledgement or retransmit segments unnecessarily. Selective ACK (SACK) can reduce the number of retransmitted segments and increase throughput under some circumstances.
	timestamp	Measures the round-trip time (RTT) of a TCP segment between two nodes on a network. Time stamps are always sent and echoed in both directions.
	window-scale	Allows the ACE to use a window-scale factor that increases the size of the TCP send and receive buffers. The sender specifies a window-scale factor in a SYN segment that determines the send and receive window size for the duration of the connection.
	clear	Clears the specified option from any segment that has it set and allows the segment. This is the default action on the explicitly supported options.

Command Modes
Parameter map connection configuration mode
Admin and user contexts

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines

Using the **tcp-options** command, the ACE permits you to allow or clear the following explicitly supported TCP options specified in a SYN segment:

- Selective Acknowledgement (SACK)
- Time stamp
- Window Scale

You can specify this command multiple times to configure different options and actions. If you specify the same option with different actions, the ACE uses the order of precedence to decide which action to use.

The order of precedence for the actions in this command is as follows:

1. Drop
2. Clear
3. Allow

Table 2-5 lists the TCP options not explicitly supported by the ACE.

Table 2-5 *Unsupported TCP Options*

Kind	Length	Meaning	Reference
6	6	Echo (obsoleted by option 8)	RFC 1072
7	6	Echo Reply (obsoleted by option 8)	RFC 1072
9	2	Partial Order Connection Permitted	RFC 1693
10	3	Partial Order Service Profile	RFC 1693
11		CC	RFC 1644
12		CC.NEW	RFC 1644
13		CC.ECHO	RFC 1644
14	3	TCP Alternate Checksum Request	RFC 1146
15	N	TCP Alternate Checksum Data	RFC 1146
16		Skeeter	[Knowles]
17		Bubba	[Knowles]
18	3	Trailer Checksum Option	[Subbu & Monroe]
19	18	MD5 Signature Option	RFC 2385
20		SCPS Capabilities	[Scott]
21		Selective Negative Acknowledgements (SNACK)	[Scott]
22		Record Boundaries	[Scott]
23		Corruption experienced	[Scott]

Table 2-5 *Unsupported TCP Options*

Kind	Length	Meaning	Reference
24		SNAP	[Sukonnik]
25		Unassigned (released 12/18/00)	
26		TCP Compression Filter	[Bellocin]

Table 2-6 lists the TCP options explicitly supported by the ACE.

Table 2-6 *Supported TCP Options*

Kind	Length	Meaning	Reference
0	-	End of Option List	RFC 793
1	-	No Operation	RFC 793
3	3	WSOPT—Window Scale	RFC 1323
4	2	Selective Acknowledgement (SACK) Permitted	RFC 2018
5	N	SACK	RFC 2018
8	10	Time Stamp Option (TSOPT)	RFC 1323

Examples

To allow the segment with the SACK option set, enter:

```
host1/Admin(config-parammap-conn) # tcp-options selective-ack allow
```

To reset the behavior of the ACE to the default of clearing the SACK option and allowing the segment, enter:

```
host1/Admin(config-parammap-conn) # no tcp-options selective-ack allow
```

You can specify a range of options for each action. If you specify overlapping option ranges with different actions, the ACE uses the order of precedence described in the “Usage Guidelines” section to decide which action to perform for the specified options.

For example, enter:

```
host1/Admin(config-parammap-conn) # tcp-options range 6 7 allow
host1/Admin(config-parammap-conn) # tcp-options range 9 18 clear
host1/Admin(config-parammap-conn) # tcp-options range 19 26 drop
```

To remove the TCP option ranges from the configuration, enter:

```
host1/Admin(config-parammap-conn) # no tcp-options range 6 7 allow
host1/Admin(config-parammap-conn) # no tcp-options range 9 18 clear
host1/Admin(config-parammap-conn) # no tcp-options range 19 26 drop
```

Related Commands

[show parameter-map](#)

(config-parammap-conn) urgent-flag

To set the Urgent Pointer policy, use the **urgent-flag** command. Use the **no** form of this command to return to the default setting of clearing the Urgent flag.

urgent-flag { **allow** | **clear** }

no urgent-flag

Syntax Description	allow	clear
	Permits the status of the Urgent flag. This is the default. If the Urgent flag is set, the offset in the Urgent Pointer that indicates the location of the urgent data is valid. If the Urgent flag is not set, the offset in the Urgent Pointer is invalid.	Sets the Urgent flag to 0, which invalidates the offset in the Urgent Pointer.

Command Modes	Parameter map connection configuration mode Admin and user contexts
---------------	--

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines	<p>If the Urgent control bit (flag) is set in the TCP header, it indicates that the Urgent Pointer is valid. The Urgent Pointer contains an offset that indicates the location of the segment that follows the urgent data in the payload. Urgent data is data that should be processed as soon as possible, even before normal data is processed. The ACE permits you to allow or clear the Urgent flag. If you clear the Urgent flag, you invalidate the Urgent Pointer.</p> <p>The ACE clears the Urgent flag for any traffic above Layer 4. If you have enabled server connection reuse (see the <i>Cisco 4700 Series Application Control Engine Appliance Security Configuration Guide</i>), the ACE does not pass the Urgent flag value to the server.</p>
------------------	--

Examples	<p>To clear the Urgent flag, enter:</p> <pre>host1/Admin(config-parammap-conn)# urgent-flag clear</pre> <p>To reset the ACE to its default of allowing the Urgent flag, enter:</p> <pre>host1/Admin(config-parammap-conn)# no urgent-flag</pre>
----------	---

Related Commands	show parameter-map
------------------	------------------------------------

Parameter Map HTTP Configuration Mode Commands

Parameter map HTTP configuration mode commands allow you to specify an HTTP-type parameter map and define its settings. To create an HTTP-type parameter map and access parameter map HTTP configuration mode, use the **parameter-map type http** command in configuration mode. The prompt changes to (config-parammap-http). Use the **no** form of the command to remove an HTTP-type parameter map from the configuration.

```
parameter-map type http name
```

```
no parameter-map type http name
```

Syntax Description

<i>name</i>	Name assigned to the parameter map. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.
-------------	---

Command Modes

Configuration mode
Admin and user contexts

Command History

Release	Modification
A1(7)	This command was introduced.

Usage Guidelines

The commands in this mode require the connection feature in your user role. For details about role-based access control (RBAC) and user roles, see the *Cisco 4700 Series Application Control Engine Appliance Virtualization Configuration Guide*.

After you create and configure a parameter map, you must associate the parameter map with a policy map to activate it. For details, see the [\(config-pmap-c\) appl-parameter http advanced-options](#) command in the “[Policy Map Configuration Mode Commands](#)” section.

Examples

To create an HTTP-type parameter map called HTTP_MAP, enter:

```
host1/Admin(config)# parameter-map type http HTTP_MAP
host1/Admin(config-parammap-http)#
```

Related Commands

[\(config\) parameter-map type](#)
[\(config-pmap-c\) appl-parameter http advanced-options](#)
[show parameter-map](#)

(config-parammap-http) case-insensitive

To enable case-insensitive matching for HTTP matching only, use the **case-insensitive** command. With case-insensitive matching enabled, uppercase and lowercase letters are considered the same. By default, the ACE CLI is case sensitive. Use the **no** form of this command to reset the ACE to its default of case-sensitive HTTP matching.

case-insensitive

no case-insensitive

Syntax Description This command has no keywords or arguments.

Command Modes Parameter map HTTP configuration mode
Admin and user contexts

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines When enabled, case insensitivity applies to the following:

- HTTP header names and values
- HTTP cookie names and values
- URL strings
- HTTP deep inspection

Examples To enable case-insensitive-matching, enter:

```
host1/Admin(config-parammap-http)# case-insensitive
```

To reenable case-sensitive matching, enter:

```
host1/Admin(config-parammap-http)# no case-insensitive
```

Related Commands [show parameter-map](#)

(config-parammap-http) compress

To define the parameters that the ACE uses when compressing HTTP traffic, use the **compress** command. Use the **no** form of this command to remove the HTTP compression.

```
compress { mimetype type/subtype | minimum-size size | user-agent string }
```

```
no compress { mimetype type/subtype | minimum-size size | user-agent string }
```

Syntax Description		
mimetype <i>type/subtype</i>		Specifies the Multipurpose Internet Mail Extension (MIME) type to compress. The default is text/* which includes all text MIME types, such as text/html, text/plain, and so on.
minimum-size <i>size</i>		Specifies the threshold at which compression occurs. The ACE compresses files that are the specified minimum size or larger. The default is 512 bytes.
user-agent <i>string</i>		Specifies the text string in the request to match. The ACE does not compress the response to a request when the request contains the specified user agent string. The default is none.

Command Modes	
	Parameter map HTTP configuration mode Admin and user contexts

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines	
	This command has no usage guidelines.

Examples	
	To specify compression of all image MIME types, enter: <pre>host1/Admin(config-parammap-http)# compress mimetype image/*</pre>
	To specify the user agent string .*Konqueror.*, enter: <pre>host1/Admin(config-parammap-http)# compress user-agent .*Konqueror.*</pre>

Related Commands	
	(config-pmap-lb-c) compress

(config-parammap-http) length-exceed

To configure how the ACE handles URLs or cookies that exceed the maximum parse length, use the **length** command. Use the **no** form of this command to reset the ACE to its default of stopping load balancing and discarding a packet when its URL or cookie exceeds the maximum parse length.

length-exceed { **continue** | **drop** }

no length-exceed

Syntax Description	continue	drop
	Specifies that the ACE continue load balancing when the maximum parse length is exceeded.	Specifies that the ACE stop load balancing when the maximum parse length is exceeded. This is the default.

Command Modes
Parameter map HTTP configuration mode
Admin and user contexts

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines
When you specify the **continue** keyword, the [\(config-parammap-http\) persistence-rebalance](#) command is disabled if the total length of all cookies, HTTP headers, and URLs exceeds the maximum parse-length value.

Examples
To continue load balancing when the maximum parse length is exceeded, enter:
host1/Admin(config-parammap-http) # **length-exceed continue**

To reset the ACE to its default of stopping load balancing and discarding a packet when its URL or cookie exceeds the maximum parse length, enter:

```
host1/Admin(config-parammap-http) # no length-exceed
```

Related Commands
[show parameter-map](#)
[\(config-parammap-http\) persistence-rebalance](#)

(config-parammap-http) persistence-rebalance

To enable the ACE to send a GET request to the real server that was used for the last GET request, use the **persistence-rebalance** command. By default, HTTP persistence is disabled. Use the **no** form of this command to reset persistence to the default setting of disabled.

persistence-rebalance

no persistence-rebalance

Syntax Description This command has no keywords or arguments.

Command Modes Parameter map HTTP configuration mode
Admin and user contexts

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines With persistence rebalance enabled, when successive GET requests result in load balancing that chooses the same policy, the ACE sends the request to the real server used for the last GET request. This behavior prevents the ACE from load balancing every request and recreating the server-side connection on every GET request, producing less overhead and better performance.

Another effect of persistence rebalance is that header insertion and cookie insertion, if enabled, occur for every request instead of only the first request.

If a real server is enabled with the NTLM Microsoft authentication protocol, we recommend that you leave persistence rebalance disabled. NTLM is a security measure that is used to perform authentication with Microsoft remote access protocols. When a real server is enabled with NTLM, every connection to the real server must be authenticated; typically, each client user will see a pop-up window prompting for a username and password. Once the connection is authenticated, all subsequent requests on the same connection will not be challenged. However, when the server load balancing function is enabled and configured with persistence rebalance, a subsequent request may point to a different real server causing a new authentication handshake.

Examples To enable persistence rebalance, enter:

```
host1/Admin(config-parammap-http)# persistence-rebalance
```

To reset persistence rebalance to the default setting of disabled, enter:

```
host1/Admin(config-parammap-http)# no persistence-rebalance
```

Related Commands [show parameter-map](#)
[\(config-pmap-lb-c\) insert-http](#)
[\(config-sticky-cookie\) cookie insert](#)

(config-parammap-http) server-conn reuse

To configure TCP server reuse, use the **server-conn reuse** command. TCP server reuse allows the ACE to reduce the number of open connections on a server by allowing connections to persist and be reused by multiple client connections. Use the **no** form of this command to disable TCP server reuse.

server-conn reuse

no server-conn reuse

Syntax Description

This command has no keywords or arguments.

Command Modes

Parameter map HTTP configuration mode

Admin and user contexts

Command History

Release	Modification
A1(7)	This command was introduced.

Usage Guidelines

The ACE maintains a pool of TCP connections that can be reused if the client connection and the server connection share the same TCP options. For information about how the ACE handles TCP options, see the *Cisco 4700 Series Application Control Engine Appliance Security Configuration Guide*. For proper operation of this feature, follow these TCP server reuse configuration recommendations and restrictions:

- Ensure that the ACE maximum segment size (MSS) is the same as the server MSS.
- Configure Port Address Translation (PAT) on the interface that is connected to the real server. PAT prevents collisions when a client stops using a server connection and then that connection is reused by another client. Without PAT, if the original client tries to reuse the original server connection, it is no longer available. For details about configuring PAT, see the *Cisco 4700 Series Application Control Engine Appliance Security Configuration Guide*.
- Configure the same TCP options that exist on the TCP server.
- Ensure that all real servers within a server farm have identical configurations.

Another effect of TCP server reuse is that header insertion and cookie insertion, if enabled, occur for every request instead of only the first request.

Examples

To enable TCP server reuse, enter:

```
host1/Admin(config-parammap-http) # server-conn reuse
```

To disable TCP server reuse, enter:

```
host1/Admin(config-parammap-http) # no server-conn reuse
```

Related Commands

[show parameter-map](#)
[\(config-parammap-http\) persistence-rebalance](#)
[\(config-pmap-lb-c\) insert-http](#)
[\(config-sticky-cookie\) cookie insert](#)

(config-parammap-http) set content-maxparse-length

To set the maximum number of bytes to parse in HTTP content, use the **set content-maxparse-length** command. Use the **no** form of this command to reset the maximum parse length to the default of 4096 bytes.

set content-maxparse-length *bytes*

no set content maxparse-length

Syntax Description

<i>bytes</i>	Maximum number of bytes to parse in HTTP content. Enter an integer from 1 to 65535. The default is 4096 bytes.
--------------	--

Command Modes

Parameter map HTTP configuration mode
Admin and user contexts

Command History

Release	Modification
A1(7)	This command was introduced.

Usage Guidelines

This command has no usage guidelines.

Examples

To set the maximum parse length to 8192, enter:

```
host1/Admin(config-parammap-http)# set content-maxparse-length 8192
```

To reset the maximum parse length to the default of 4096 bytes, enter:

```
host1/Admin(config-parammap-http)# no set content-maxparse-length
```

Related Commands

[show parameter-map](#)

(config-parammap-http) set header-maxparse-length

To set the maximum number of bytes to parse for cookies, HTTP headers, and URLs, use the **set header-maxparse-length** command. Use the **no** form of this command to reset the HTTP header maximum parse length to the default of 2048 bytes.

set header-maxparse-length *bytes*

no set-header maxparse-length

Syntax Description	<i>bytes</i>	Maximum number of bytes to parse for the total length of all cookies, HTTP headers, and URLs. Enter an integer from 1 to 65535. The default is 2048 bytes.
---------------------------	--------------	--

Command Modes	Parameter map HTTP configuration mode Admin and user contexts
----------------------	--

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines	This command has no usage guidelines.
-------------------------	---------------------------------------

Examples	To set the HTTP header maximum parse length to 8192, enter: host1/Admin(config-parammap-http)# set header-maxparse-length 8192
	To reset the HTTP header maximum parse length to the default of 2048 bytes, enter: host1/Admin(config-parammap-http)# no set header-maxparse-length

Related Commands	show parameter-map
-------------------------	------------------------------------

(config-parammap-http) set secondary-cookie-delimiters

To define a list of ASCII-character delimiter strings that you can use to separate the cookies in a URL string, use the **set secondary-cookie-delimiters** command. Use the **no** form of this command to reset the delimiter string list to the default of `/?&#+`.

```
set secondary-cookie-delimiters text
```

```
no set secondary-cookie-delimiters
```

Syntax Description	<i>text</i>	Delimiter string. Enter an unquoted text string with no spaces and a maximum of four characters. The order of the delimiters in the list does not matter. The default list of delimiters is <code>/?&#+</code> .
---------------------------	-------------	--

Command Modes	Parameter map HTTP configuration mode Admin and user contexts
----------------------	--

Command History	<table border="1"> <thead> <tr> <th>Release</th> <th>Modification</th> </tr> </thead> <tbody> <tr> <td>A1(7)</td> <td>This command was introduced.</td> </tr> </tbody> </table>	Release	Modification	A1(7)	This command was introduced.
Release	Modification				
A1(7)	This command was introduced.				

Usage Guidelines	<p>Cookies and their delimiters appear in GET request lines. In the following example of a GET request line, the ampersand (&) that appears between name-value pairs is the secondary cookie delimiter. The question mark (?) begins the URL query and is not configurable.</p>
-------------------------	---

```
GET /default.cgi?user=me&hello=world&id=2 HTTP/1.1
```

Examples	<p>To set the delimiter string list to the characters <code>!@#</code>, enter:</p> <pre>host1/Admin(config-parammap-http)# set secondary-cookie-delimiters !@#</pre>
-----------------	--

To reset the delimiter string list to the default of `/?&#+`, enter:

```
host1/Admin(config-parammap-http)# no set secondary-cookie-delimiters
```

Related Commands	show parameter-map
-------------------------	------------------------------------

Parameter Map Optimization Configuration Mode Commands

Parameter map optimization configuration mode commands allow you to create an optimization HTTP-type parameter map and define its application acceleration settings. To create an optimization HTTP-type parameter map and access parameter map optimization configuration mode, use the **parameter-map type optimization http** command in configuration mode. The prompt changes to (config-parammap-optmz). Use the **no** form of the command to remove an optimization HTTP-type parameter map from the configuration.

```
parameter-map type optimization http map_name
```

```
no parameter-map type optimization http map_name
```

Syntax Description

<i>map_name</i>	Enter a unique name as an unquoted text string with a maximum of 64 alphanumeric characters.
-----------------	--

Command Modes

Configuration mode
Admin and user contexts

Command History

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 *Cisco 4700 Series Application Control Engine Appliance Virtualization Configuration Guide*.

An optimization HTTP parameter map can be optionally specified in an optimization HTTP policy map to identify the association between an optimization HTTP action list and the parameter map. The optimization HTTP action list defines what to do, while the optimization HTTP parameter map defines the specific details about how to accomplish the application acceleration action. For details, see the “[Policy Map Optimization Configuration Mode Commands](#)” section.

Examples

To create an optimization HTTP-type parameter map, enter:

```
host1/Admin(config)# parameter-map type optimization http OPTIMIZE_PARAM_MAP
host1/Admin(config-parammap-optmz)#
```

To remove a Layer 7 optimization parameter map from the configuration, enter:

```
host1/Admin(config)# no parameter-map type optimization http OPTIMIZE_PARAM_MAP
```

Related Commands

[\(config\) parameter-map type](#)
[\(config\) action-list type](#)
[show parameter-map](#)

(config-parammap-optmz) appscope optimize-rate-percent

To control the AppScope features that measure application acceleration performance by the optional Cisco AVS 3180A Management Station, use the **appscope optimize-rate-percent** command. Use the **no** form of the command to revert to the default AppScope performance rate settings.

appscope optimize-rate-percent *value* **passthru-rate-percent** *value*

no appscope optimize-rate-percent *value* **passthru-rate-percent** *value*

Syntax Description

<i>value</i>	Percentage of all requests (or sessions) to be sampled for performance with acceleration (optimization) applied. All applicable optimizations for the class will be performed. Valid values are from 0 to 100 percent. The default is 10 percent. This value plus the passthru-rate-percent value must not exceed 100.
passthru-rate-percent <i>value</i>	Percentage of all requests (or sessions) to be sampled for performance without optimization. No optimizations for the class will be performed. Valid values are from 0 to 100 percent. The default is 10 percent. This value plus the optimize-rate-percent value must not exceed 100.

Command Modes

Parameter map optimization configuration mode
Admin and user contexts

Command History

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 action list optimization configuration mode. See the “[Action List 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

To specify a percentage of all requests (or sessions) to be sampled for performance with acceleration and without optimization applied by AppScope, enter:

```
host1/Admin(config-parammap-optmz)# appscope optimize-rate-percent 50  
passthru-rate-percent 50
```

To revert to the default rate AppScope performance rate settings of 10 percent, enter:

```
host1/Admin(config-parammap-optmz)# no appscope optimize-rate-percent 50  
passthru-rate-percent 50
```

Related Commands

([config-actlist-optm](#)) **appscope**
([config-parammap-optmz](#)) **request-grouping-string**

(config-parammap-optmz) basefile anonymous-level

To define the base file anonymity level for the all-user delta optimization method, use the **basefile anonymous-level** command. By default, the base file anonymity level is disabled. Use the **no** form of the command to revert to the default base file anonymity level of 0.

basefile anonymous-level *value*

no basefile anonymous-level *value*

Syntax Description

<i>value</i>	Base file anonymity level for the all-user delta optimization method. Valid values are from 0 to 50. The default is a value of 0 (disables anonymity).
--------------	--

Command Modes

Parameter map optimization configuration mode
Admin and user contexts

Command History

Release	Modification
A1(7)	This command was introduced.

Usage Guidelines

The string can contain a URL regular expression that defines a set of URLs in which URLs that differ only by their query parameters are to be treated as separate URLs in AppScope reports.

Typically, in an AppScope report organized by URL, matching URLs that differ only in their query parameters are treated as the same URL and are not listed on separate lines. Use the **request-grouping-string** command to specify that all URL variations that are based on query parameters are to be treated as separate URLs for reporting purposes. Each variation will appear on a separate line in the report.

For details about the optional Cisco AVS 3180A Management Station 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*.

Examples

To specify a base file anonymity level of 25, enter:

```
host1/Admin(config-parammap-optmz)# basefile anonymous-level 25
```

To revert to the default base file anonymity level of 0, enter:

```
host1/Admin(config-parammap-optmz)# no basefile anonymous-level
```

Related Commands

[\(config-parammap-optmz\) canonical-url](#)
[\(config-parammap-optmz\) delta](#)

(config-parammap-optmz) cache key-modifier

To modify the canonical form of a URL, which is the portion before the question mark (?), to form the cache key, use the **cache key-modifier** command. This command specifies a regular expression that contains embedded variables that are expanded by the ACE. Use the **no** form of the command to remove a cache key modifier.

cache key-modifier {*string parameter_expander_function*}

no cache key-modifier {*regular_expression parameter_expander_function*}

Syntax Description

<i>string</i>	A regular expression. Enter an unquoted text string with no spaces and a maximum of 255 alphanumeric characters. Alternatively, you can enter a text string with spaces provided that you enclose the entire string in quotation marks (“”). The ACE supports the use of regular expressions for matching string expressions. The “Usage Guidelines” section lists the supported characters that you can use for matching string expressions.
<i>parameter_expander_function</i>	A parameter expander function that evaluate to strings. The “Usage Guidelines” section lists the parameter expander functions that you can use.

Command Modes

Parameter map optimization configuration mode
 Admin and user contexts

Command History

Release	Modification
A1(7)	This command was introduced.

Usage Guidelines

The key that the ACE uses for any given requesting URL comprises one or more of the following two components:

- Query parameters—The URL portion after a question mark (?). You can modify query parameters by using the **cache parameter** command, which can be used to include selected query parameters, a cookie value, an HTTP header value, or other values.
- Canonical URL—The URL portion up to a question mark (?). You can modify the canonical URL by using the **cache key-modifier** command.

The expanded string that results from the **cache key-modifier** command replaces the default canonical URL portion of the cache key. If you do not specify the **cache key-modifier** command, the canonical URL is used as the default value for the URL portion of the cache key (there may also be a query parameter portion).

For details on modifying the cache key, see the *Cisco 4700 Series Application Control Engine Appliance Application Acceleration and Optimization Configuration Guide*.

The following table lists the supported characters that you can use for matching string expressions.

Convention	Description
.	One of any character.
.*	Zero or more of any character.
\.	Period (escaped).
[charset]	Match any single character from the range.
[^charset]	Do not match any character in the range. All other characters represent themselves.
()	Expression grouping.
(expr1 expr2)	OR of expressions.
(expr)*	0 or more of expression.
(expr)+	1 or more of expression.
expr{m,n}	Repeat the expression between <i>m</i> and <i>n</i> times, where <i>m</i> and <i>n</i> have a range of 1 to 255.
expr{m}	Match the expression exactly <i>m</i> times. The range for <i>m</i> is from 1 to 255.
expr{m,}	Match the expression <i>m</i> or more times. The range for <i>m</i> is from 1 to 255.
\a	Alert (ASCII 7).
\b	Backspace (ASCII 8).
\f	Form-feed (ASCII 12).
\n	New line (ascii 10).
\r	Carriage return (ASCII 13).
\t	Tab (ASCII 9).
\v	Vertical tab (ASCII 11).
\0	Null (ASCII 0).
\\	Backslash.
\x##	Any ASCII character as specified in two-digit hexadecimal notation.

The following table lists the parameter expander functions that you can use.

Variable	Description
<p><code>\$ (number)</code></p>	<p>Expands to the corresponding matching subexpression (by <i>number</i>) in the URL pattern. Subexpressions are marked in a URL pattern using parentheses (). The numbering of the subexpressions begins with 1 and is the number of the left-parenthesis “(“ counting from the left. You can specify any positive integer for the number. <code>\$(0)</code> matches the entire URL. For example, if the URL pattern is <code>((http://server/.*)/(.*)/a.jsp)</code>, and the URL that matched it is the following:</p> <p><code>http://server/main/sub/a.jsp?category=shoes&session=99999</code>, then the following are correct:</p> <p><code>\$(0) = http://server/main/sub/a.jsp</code> <code>\$(1) = http://server/main/sub/</code> <code>\$(2) = http://server/main</code> <code>\$(3) = sub</code></p> <p>If the specified subexpression does not exist in the URL pattern, then the variable expands to the empty string.</p>
<p><code>\$http_query_string()</code></p>	<p>Expands to the value of the whole query string in the URL. For example, if the URL is</p> <p><code>http://myhost/dohis?param1=value1&param2=value2</code></p> <p>then the following is correct:</p> <p><code>\$http_query_string() = param1=value1&param2=value2</code></p> <p>This function applies to both GET and POST requests.</p>
<p><code>\$http_query_param (query-param-name)</code></p> <p>this obsolete syntax is also supported:</p> <p><code>\$param (query-param-name)</code></p>	<p>Expands to the value of the named query parameter (case sensitive). For example, if the URL is</p> <p><code>http://server/main/sub/a.jsp?category=shoes&session=99999</code></p> <p>then the following are correct:</p> <p><code>\$http_query_param(category) = shoes</code> <code>\$http_query_param(session) = 99999</code></p> <p>If the specified parameter does not exist in the query, then the variable expands to the empty string. This function applies to both GET and POST requests.</p>
<p><code>\$http_cookie (cookie-name)</code></p>	<p>Evaluates to the value of the named cookie. For example, <code>\$http_cookie(cookiexyz)</code>. The cookie name is case sensitive.</p>
<p><code>\$http_header (request-header-name)</code></p>	<p>Evaluates to the value of the specified HTTP request header. In the case of multivalued headers, it is the single representation as specified in the HTTP specification. For example, <code>\$http_header(user-agent)</code>. The HTTP header name is not case sensitive.</p>

Variable	Description
\$http_method()	Evaluates to the HTTP method used for the request, such as GET or POST.
Boolean Functions: \$http_query_param_present (<i>query-param-name</i>) \$http_query_param_notpresent (<i>query-param-name</i>) \$http_cookie_present (<i>cookie-name</i>) \$http_cookie_notpresent (<i>cookie-name</i>) \$http_header_present (<i>request-header-name</i>) \$http_header_notpresent (<i>request-header-name</i>) \$http_method_present (<i>method-name</i>) \$http_method_notpresent (<i>method-name</i>)	Evaluates to a Boolean value: True or False, depending on the presence or absence of the element in the request. The elements are a specific query parameter (<i>query-param-name</i>), a specific cookie (<i>cookie-name</i>), a specific request header (<i>request-header-name</i>), or a specific HTTP method (<i>method-name</i>). All identifiers are case sensitive except for the HTTP request header name.

Examples

For example, enter:

```
host1/Admin(config-parammap-optmz)# cache key-modifier $http://www(1)
```

To remove a cache key modifier, enter:

```
host1/Admin(config-parammap-optmz)# no cache key-modifier
```

Related Commands

[\(config-parammap-optmz\) cache parameter](#)
[\(config-parammap-optmz\) cache ttl](#)

(config-parammap-optmz) cache parameter

To modify the query parameter part of a URL, which is the portion after the question mark (?), to form the cache key, use the **cache parameter** command. Use the **no** form of the command to remove a cache parameter.

cache parameter *parameter_expander_function*

no cache parameter *parameter_expander_function*

Syntax Description

parameter_expander_function Parameter expander function that evaluates to strings. Use the forwardslash (/) character when combining multiple parameter expander functions (for example, **cache parameter \$http_cookie(ID)/\$http_query_param(category)**). The maximum string value is 255 characters. See the “(config-parammap-optmz) cache key-modifier” section for a listing of the parameter expander functions that you can use.

Command Modes Parameter map optimization configuration mode
Admin and user contexts

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines The key that the ACE uses for any given requesting URL comprises one or more of the following two components:

- Query parameters—The URL portion after a question mark (?). You can modify query parameters by using the **cache parameter** command, which can be used to include selected query parameters, a cookie value, an HTTP header value, or other values.
- Canonical URL—The URL portion up to a question mark (?). You can modify the canonical URL by using the **cache key-modifier** command.

The **cache parameter** command specifies an expression that includes one or more parameter expander functions if you want to modify the parameter portion of the cache key. This command specifies one or more parameter expander functions that evaluate to strings. These strings are appended to the canonical URL to form the last portion of the cache key. The parameter expander functions are listed in the [\(config-parammap-optmz\) cache key-modifier](#) command.

The string specified in the **cache parameter** command replaces the default query parameter that is used in the cache key. If you do not specify the **cache parameter** command, the query parameter portion of the URL is used as the default value for this portion of the cache key. The canonical URL, possibly modified by the **cache key-modifier** command, is the first part of the cache key.

For details on modifying the cache key, see the *Cisco 4700 Series Application Control Engine Appliance Application Acceleration and Optimization Configuration Guide*.

Examples To set the value of the query parameter portion of the cache key, enter:

```
host1/Admin(config-parammap-optmz)# cache parameter $http_query_param (version)
```

To remove a cache parameter, enter:

```
host1/Admin(config-parammap-optmz)# no cache parameter
```

Related Commands [\(config-parammap-optmz\) cache key-modifier](#)
[\(config-parammap-optmz\) cache ttl](#)

(config-parammap-optmz) cache ttl

To define the ACE cache freshness settings, use the **cache ttl** command. Use the **no** form of the command to revert to a default cache time-to-live value.

```
cache ttl {min time | max time | percent value}
```

```
no cache ttl {min time | max time | percent value}
```

Syntax Description

min time	Minimum time in seconds that an object without an explicit expiration time should be considered fresh. The min keyword specifies the minimum time that the content can be cached for, which corresponds to the time-to-live value of the content. In the case of a new item that is valid for three hours, this value would be 3 x 60 x 60 = 10800 seconds. If you perform static caching (the flashforward-object action), this value should normally be 0. If you perform dynamic caching (the cache dynamic action) this value should be set to indicate how long the ACE should cache the page. Valid values are from 0 to 2147483647 seconds. The default is 0.
max time	Maximum time in seconds than an object without an explicit expiration time should be considered fresh. The max keyword determines how the ACE handles the case when the object has passed its cache minimum time-to-live value. Valid values are from 0 to 2147483647 seconds. The default is 300 seconds.
percent value	Percent of an object's age at which an embedded object without an explicit expiration time is considered fresh. Valid values are from 0 to 100 percent. The default is 0 percent.

Command Modes

Parameter map optimization configuration mode
Admin and user contexts

Command History

Release	Modification
A1(7)	This command was introduced.

Usage Guidelines

This command sets the maximum time (**max** keyword) or the minimum time (**min** keyword) in seconds that an object without an explicit expiration time should be considered fresh. The **percent** keyword sets the percent of an object's age at which an embedded object without an explicit expiration time is considered fresh.

Examples

To specify a minimum time-to-live value of 1000 seconds in which the content can be cached, enter:

```
host1/Admin(config-parammap-optmz)# cache ttl min 1000
```

To revert to a default cache time-to-live value, enter:

```
host1/Admin(config-parammap-optmz)# no cache ttl min
```

Related Commands

[\(config-parammap-optmz\) cache key-modifier](#)

[\(config-parammap-optmz\) cache parameter](#)

[\(config-parammap-optmz\) cache ttl](#)

(config-parammap-optmz) cache-policy request

To override client request headers (primarily for embedded objects), use the **cache-policy request** command. Use the **no** form of the command to remove a cache policy request selection.

```
cache-policy request {override-all | override-cache-ctl-no-cache}
```

```
no cache-policy request {override-all | override-cache-ctl-no-cache}
```

Syntax Description

override-all	Specifies that all cache request headers are ignored.
override-cache-ctl-no-cache	Overrides the Cache-Control: no cache HTTP header from a request. This keyword is used for a flashforward-object command action (see the “ (config-actlist-optm) flashforward-object ” section). Typically, if there is a cache control request header stating no cache, the ACE will not cache this object. The override-cache-ctl-no-cache keyword instructs the ACE to ignore the Cache-Control: no cache header from the request side.

Command Modes

Parameter map optimization configuration mode

Admin and user contexts

Command History

Release	Modification
A1(7)	This command was introduced.

Usage Guidelines

This command has no usage guidelines.

Examples

To instruct the ACE that all cache request headers are ignored, enter:

```
host1/Admin(config-parammap-optmz)# cache-policy request override-all
```

To remove a cache policy request selection, enter:

```
host1/Admin(config-parammap-optmz)# no cache-policy request override-all
```

Related Commands [\(config-actlist-optm\) flashforward-object](#)

(config-parammap-optmz) cache-policy response

To override origin server response headers (primarily for embedded objects), use the **cache-policy response** command. Use the **no** form of the command to remove a cache policy response selection.

cache-policy response { **override-all** | **override-cache-ctl-private** }

no cache-policy response { **override-all** | **override-cache-ctl-private** }

Syntax Description

override-all	Specifies that all cache response headers are ignored.
override-cache-ctl-private	Overrides the Cache-Control: private HTTP header from a response. This keyword is used for a flashforward-object command action (see the “ (config-actlist-optm) flashforward-object ” section) and is equivalent to static object caching. Typically, if there is a cache control response header stating private, these response headers will make the object not cacheable. The override-cache-ctl-private keyword instructs the ACE to ignore the Cache-Control: private HTTP header from a response.

Command Modes

Parameter map optimization configuration mode
Admin and user contexts

Command History

Release	Modification
A1(7)	This command was introduced.

Usage Guidelines

This command has no usage guidelines.

Examples

To instruct the ACE that all cache response headers are ignored, enter:

```
host1/Admin(config-parammap-optmz)# cache-policy response override-all
```

To remove a cache policy response selection, enter:

```
host1/Admin(config-parammap-optmz)# no cache-policy response override-all
```

Related Commands [\(config-actlist-optm\) flashforward-object](#)

(config-parammap-optmz) canonical-url

To specify a string containing a canonical URL regular expression that defines a set of URLs to which the parameter map applies, use the **canonical-url** command. Use the **no** form of the command to delete the string that contains a canonical URL regular expression.

canonical-url {*parameter-expander-function*}

no canonical-url {*parameter-expander-function*}

Syntax Description

parameter-expander-function Parameter expander function that evaluates to strings. See the “(config-parammap-optmz) cache key-modifier” section for a listing of the parameter expander functions that you can use.

Command Modes

Parameter map optimization configuration mode
Admin and user contexts

Command History

Release	Modification
A1(7)	This command was introduced.

Usage Guidelines

At least one URL must be specified using the **canonical-url** command.

Use the canonical URL function in a parameter map to specify a base file selection policy. The canonical URL function specifies a regular expression that is used to match a variety of actual URLs. All matched URLs share a single base file.

The ACE uses the canonical URL feature to modify a parameterized request to eliminate the question mark (?) and the characters that follow to identify the general part of the URL. This general URL is then used to create the base file. The ACE uses this feature to map multiple parameterized URLs to a single canonical URL.

Examples

To specify a string that contains a canonical URL regular expression, enter:

```
host1/Admin(config-parammap-optmz)# canonical-url (1)/http_query_param(category)
```

To delete the string that contains a canonical URL regular expression, enter:

```
host1/Admin(config-parammap-optmz)# no canonical-url
```

Related Commands

(config-parammap-optmz) [basefile anonymous-level](#)
(config-parammap-optmz) [cache key-modifier](#)
(config-parammap-optmz) [cache parameter](#)
(config-parammap-optmz) [expires-setting](#)

(config-parammap-optmz) clientscript-default

To configure the ACE to recognize the scripting language used on delta optimized content pages, either JavaScript or Visual Basic, use the **clientscript-default** command. Use the **no** form of the command to revert to the default JavaScript scripting language.

```
clientscript-default {javascript | vbscript}
```

```
no clientscript-default {javascript | vbscript}
```

Syntax Description

javascript	Sets the default scripting language to JavaScript (default).
vbscript	Sets the default scripting language to Visual Basic.

Command Modes

Parameter map optimization configuration mode
Admin and user contexts

Command History

Release	Modification
A1(7)	This command was introduced.

Usage Guidelines

This command has no usage guidelines.

Examples

To set the default scripting language to Visual Basic, enter:

```
host1/Admin(config-parammap-optmz)# clientscript-default vbscript
```

To revert to the default JavaScript scripting language, enter:

```
host1/Admin(config-parammap-optmz)# no clientscript-default vbscript
```

Related Commands

This command has no related commands.

(config-parammap-optmz) delta

To control the delta optimization mode used by the ACE and to configure the delta optimization operating parameters on the ACE, use the **delta** command. Use the **no** form of the command to revert to the default all-user delta optimization mode.

```
delta {all-user | cacheable-content | exclude {iframes | mime-type mime-type | non-ascii |
scripts} | first-visit | page-size {min value | max value} | per-user }
```

```
no delta {all-user | cacheable-content | exclude {iframes | mime-type mime-type | non-ascii |
scripts} | first-visit | page-size {min value | max value} | per-user }
```

Syntax Description	
all-user	Specifies the corresponding URLs are to be delta optimized using the all-user delta optimization mode. This is the default.
cacheable-content	Enables delta optimization of cacheable content. Typically, the ACE detects cacheable content and prevents its delta optimization.
exclude	Defines the cacheable objects that should not be delta optimized.
iframes	Specifies that IFrames should not be delta optimized.
mime-type <i>mime-type</i>	Specifies the Multipurpose Internet Mail Extension (MIME)-type messages that should not be delta optimized (such as image/Jpeg, text/html, application/msword, audio/mpeg). The following lists the supported mime-types: <ul style="list-style-type: none"> • application/msexcel • application/mspowerpoint • application/msword • application/octet-stream • application/pdf • application/postscript • application/x-gzip • application/x-java-archive • application/x-java-vm • application/x-messenger • application/zip • audio/* • audio/basic • audio/midi • audio/mpeg • audio/x-adpcm • audio/x-aiff • audio/x-ogg • audio/x-wav

	<ul style="list-style-type: none"> • image/* • image/gif • image/jpeg • image/png • image/tiff • image/x-3dsimage/x-bitmap • image/x-niff • image/x-portable-bitmap • image/x-portable-greymap • image/x-xpm • text/* text/sgml • text/xmcd • text/xml • video/* • video/flc • video/mpeg • video/quicktime • video/sgi • video/x-fli
non-ascii	Specifies that non-ASCII data should not be delta optimized. Specify this keyword if the content has UTF8 characters. Using this keyword excludes such UTF8 characters from delta optimization but the remainder of that page can still have delta optimization.
scripts	Specifies that JavaScript should not to be delta optimized.
first-visit	Enables delta optimization on the first visit to a web page.
page-size	Sets the minimum and maximum page size, in bytes, that can be delta optimized.
min value	Specifies the minimum page size, in bytes, that can be delta optimized. Valid values are from 1 to 250000 bytes. The default is 1024 bytes.
max value	Specifies the maximum page size, in bytes, that can be delta optimized. Valid values are 1024 to 250000 bytes. The default is 250000 bytes.
per-user	Specifies the corresponding URLs are to be delta optimized using the per-user delta optimization mode.

Command Modes

Parameter map optimization configuration mode

Admin and user contexts

Command History

Release	Modification
A1(7)	This command was introduced.

Usage Guidelines

Delta optimization mode specifies whether the web pages to be delta optimized are common to all users or personalized for individual users, which determines what kind of page deltas are generated by the ACE.

The ACE supports two delta optimization modes:

- All-user mode
- Per-user mode

In the all-user delta optimization mode, the delta is generated against a single base file that is shared by all users of the URL. The all-user delta optimization mode is usable in most cases, even in the case of dynamic personalized content if the structure of a page is common across users. The disk space overhead is minimal (the disk space requirements are determined by the number of delta optimized pages, not the number of users).

In the per-user delta optimization mode, when a specific user requests a URL, the delta for the response is generated against a base file that is created specifically for that user. The per-user delta optimization mode is useful in situations where the contents of a page (including layout elements) are different for each user. This mode delivers the highest level of delta optimization. However, a copy of the base page that is delivered to each user has to be kept in the ACE cache which increases the requirements on disk space for the ACE cache. The per-user delta optimization mode is useful for content privacy because base pages are not shared among users.

Examples

To specify that the corresponding URLs are to be delta optimized using the per-user delta optimization mode, enter:

```
host1/Admin(config-parammap-optmz)# delta per-user
```

To revert to the default all-user delta optimization mode, enter:

```
host1/Admin(config-parammap-optmz)# no delta per-user
```

To specify the MIME-type messages that should not be delta optimized, enter:

```
host1/Admin(config-parammap-optmz)# delta exclude mime-type audio/mpeg
```

To disable a delta optimization operating parameter on the ACE, enter:

```
host1/Admin(config-parammap-optmz)# no delta exclude mime-type audio/mpeg
```

Related Commands

[\(config-actlist-optm\) delta](#)
[\(config-parammap-optmz\) basefile anonymous-level](#)

((config-parammap-optmz) expires-setting

To control the period of time that objects in the client's browser remain fresh, use the **expires-setting** command. Use the **no** form of the command to remove an expiration setting.

expires-setting { **cachettl** | **time-to-live** *seconds* | **unmodified** }

no expires-setting { **cachettl** | **time-to-live** *seconds* | **unmodified** }

Syntax Description		
cachettl		Sets the freshness similar to FlashForwarded objects and uses the minimum and maximum settings configured by the cache ttl command (if set). See the “((config-parammap-optmz) cache ttl” section.
time-to-live <i>seconds</i>		The duration that objects in the client's browser remain fresh. Valid entries are from 0 to 2147483647 seconds.
unmodified		Disables browser object freshness control (default).

Command Modes Parameter map optimization configuration mode
Admin and user contexts

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines The **expires-setting** command instructs the ACE to insert an Expires response header with a time value for an object. It is not necessary to configure this command when specifying the **flashforward** command in an action list because, in this case, the ACE always inserts a long time value in the Expires header for the transformed object. The **expires-setting** command is typically used when you are not using FlashForward but want to achieve the FlashForward affect by making all of the embedded objects perceived as being fresh by the browser.

Examples To specify that the ACE use the settings configured by the **cache ttl** command, enter:

```
host1/Admin(config-parammap-optmz)# expires-setting cachettl
```

To remove an expiration setting, enter:

```
host1/Admin(config-parammap-optmz)# no expires-setting cachettl
```

Related Commands [\(\(config-parammap-optmz\) cache ttl](#)

(config-parammap-optmz) extract meta

To configure the ACE to remove HTML Meta elements from documents to prevent them from being condensed, use the **extract meta** command. By default, the ACE includes HTML Meta elements in documents. Use the **no** form of the command to include HTML Meta elements in documents.

extract meta

no extract meta

Syntax Description This command has no keywords or arguments.

Command Modes Parameter map optimization configuration mode
Admin and user contexts

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines This command has no usage guidelines.

Examples To remove HTML Meta elements from documents, enter:

```
host1/Admin(config-parammap-optmz)# extract meta
```

To include HTML Meta elements in documents, enter:

```
host1/Admin(config-parammap-optmz)# no extract meta
```

Related Commands This command has no related commands.

(config-parammap-optmz) flashforward refresh-policy

To configure the ACE to bypass FlashForward for stale embedded objects, use the **flashforward refresh-policy** command. Use the **no** form of the command to revert to the default of allowing FlashForward to indirectly refresh embedded objects.

```
flashforward refresh-policy {all | direct}
```

```
no flashforward refresh-policy {all | direct}
```

Syntax Description

all	Allows FlashForward to indirectly refresh embedded objects (default).
direct	Bypasses FlashForward for stale embedded objects so that they are directly refreshed.

Command Modes

Parameter map optimization configuration mode
Admin and user contexts

Command History

Release	Modification
A1(7)	This command was introduced.

Usage Guidelines

Request headers that the ACE sends to the origin server for stale embedded objects (indirect GET) may not be accepted by the origin server and cause errors. In this case, specify **direct** to prevent this behavior. FlashForward is disabled by default; you must enable it by specifying the following commands in action list optimization mode: **flashforward** and **flashforward-object** (for embedded objects).

Examples

To bypass FlashForward for stale embedded objects, enter:

```
host1/Admin(config-parammap-optmz)# flashforward refresh-policy direct
```

To revert to the default of allowing FlashForward to indirectly refresh embedded objects, enter:

```
host1/Admin(config-parammap-optmz)# no flashforward refresh-policy
```

Related Commands

[\(config-actlist-optm\) flashforward](#)
[\(config-actlist-optm\) flashforward-object](#)

(config-parammap-optmz) ignore-server-content

To specify a comma-separated list of HTTP response codes for which the response body must not be read (ignored), use the **ignore-server-content** command. Use the **no** form of the command to remove one or more response codes to ignore.

ignore-server-content *value*

no ignore-server-content *value*

Syntax Description	<i>value</i>	The response code as an unquoted text string with a maximum of 64 alphanumeric characters. For example, a response code value of 302 directs the ACE to ignore the response body in the case of a 302 (redirect) response from the origin server.
---------------------------	--------------	---

Command Modes	Parameter map optimization configuration mode Admin and user contexts
----------------------	--

Command History	<table border="1"> <thead> <tr> <th>Release</th> <th>Modification</th> </tr> </thead> <tbody> <tr> <td>A1(7)</td> <td>This command was introduced.</td> </tr> </tbody> </table>	Release	Modification	A1(7)	This command was introduced.
Release	Modification				
A1(7)	This command was introduced.				

Usage Guidelines	This command has no usage guidelines.
-------------------------	---------------------------------------

Examples	<p>To specify a response code value of 302 to ignore, enter:</p> <pre>host1/Admin(config-parammap-optmz)# ignore-server-content 302</pre> <p>To remove one or more response codes to ignore, enter:</p> <pre>host1/Admin(config-parammap-optmz)# no ignore-server-content</pre>
-----------------	---

Related Commands	This command has no related commands.
-------------------------	---------------------------------------

(config-parammap-optmz) parameter-summary parameter-value-limit

To set the maximum number of bytes that are logged for each parameter value in the parameter summary of a transaction log entry in the statistics log, use the **parameter-summary parameter-value-limit** command. Use the **no** form of the command to revert to the default of 100 bytes as the parameter summary value.

parameter-summary parameter-value-limit *bytes*

no parameter-summary parameter-value-limit *bytes*

Syntax Description	<i>bytes</i>	Maximum number of bytes that are logged for each parameter value in the parameter summary of a transaction log entry in the statistical log. If a parameter value is longer than this limit, it is truncated at the specified parameter limit. Valid values are from 0 to 10,000 bytes. The default is 100 bytes.
---------------------------	--------------	---

Command Modes	Parameter map optimization configuration mode Admin and user contexts
----------------------	--

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines	This command has no usage guidelines.
-------------------------	---------------------------------------

Examples	To specify 5000 bytes as the value of the parameter summary, enter: <pre>host1/Admin(config-parammap-optmz)# parameter-summary parameter-value-limit 5000</pre>
	To revert to the default of 100 bytes as the value of the parameter summary, enter: <pre>host1/Admin(config-parammap-optmz)# no parameter-summary parameter-value-limit</pre>

Related Commands	<p>(config) logging host</p> <p>(config-actlist-optm) appscope</p> <p>(config-parammap-optmz) appscope optimize-rate-percent</p> <p>(config-parammap-optmz) request-grouping-string</p>
-------------------------	---

(config-parammap-optmz) post-content-buffer-limit

To set the buffer size of an HTTP POST to a maximum number of kilobytes, use the **post-content-buffer-limit** command. Use the **no** form of the command to revert to the default buffer size of 40K.

post-content-buffer-limit *value*

no post-content-buffer-limit *value*

Syntax Description

<i>value</i>	The buffer size for POST data for the purpose of logging transaction parameters in the statistics log. Valid values are 0 to 1000 KB. The default is 40 KB. Parameters beyond this limit will not be logged by the ACE.
--------------	---

Command Modes

Parameter map optimization configuration mode
Admin and user contexts

Command History

Release	Modification
A1(7)	This command was introduced.

Usage Guidelines

An HTTP POST can send a very large (effectively unlimited) amount of data; in an extreme case, the client can keep sending a stream of data for the server to handle. In order to parse and inspect the POST data, the ACE needs to load the data into a buffer in memory.

Two types of standard HTTP form POST operations are as follows (they are distinguished by the value in the Content-Type header):

- **application/x-www-form-urlencoded**—This type represents the majority of all HTTP POSTs. This type is just a standard POST of a webpage form.
- **multipart/form-data**—This type is much less common. It allows browser users to upload files to a website or application. For example, if you use a web-based email program, and you want to attach a file to an e-mail that you are sending, the upload of the file is done using this type. Another usage (even less common) of this type of HTTP POST is to send binary data (for example, from a custom browser plug-in, or from a non-browser HTTP client).

Examples

To specify a buffer size of 1000 KB, enter:

```
host1/Admin(config-parammap-optmz)# post-content-buffer-limit 1000
```

To revert to the default buffer size of 40 KB, enter:

```
host1/Admin(config-parammap-optmz)# no post-content-buffer-limit
```

Related Commands

This command has no related commands.

(config-parammap-optmz) rebase

To control the rebasing of base files by the ACE, use the **rebase** command. Use the **no** form of the command to revert to a default rebase setting.

```
rebase { delta-percent value | flashforward-percent value | history-size value | modification-cooloff-period value | reset-period value }
```

```
no rebase { delta-percent value | flashforward-percent value | history-size value | modification-cooloff-period value | reset-period value }
```

Syntax Description

delta-percent <i>value</i>	Specifies the delta threshold at which rebasing is triggered. This number represents the size of a page delta relative to the page total size, expressed as a percentage. Valid values are from 0 to 10000 percent. The default threshold is 50 percent.
flashforward-percent <i>value</i>	Specifies a rebase, based on the percent of FlashForwarded URLs in the response. Rebasing is triggered when the difference between the percentages of FlashForwarded URLs in the delta response and the base file exceed the threshold. Valid values are from 0 to 10000 percent. The default is 50 percent. The flashforward-percent keyword provides a threshold control for rebasing based on the percent of FlashForwarded URLs in the response. Where the delta-percent keyword triggers rebasing when the delta response size exceeds the threshold as a percentage of base file size; the flashforward-percent keyword triggers rebasing when the difference between the percentages of FlashForwarded URLs in the delta response and the base file exceed the threshold.
history-size <i>value</i>	Controls how much history is stored before resetting. Once the sample collection reaches the specified history size, the ACE resets all rebase control parameters to zero and starts over. Using the history-size keyword prevents the base file from becoming too rigid. That is, if a base file has served approximately one million pages, then it would take another half million unfavorable responses before the base file can be rebased. Valid values are from 10 to 2147483647 pages. The default value for this parameter is 1000 pages.

modification-cooloff-period <i>value</i>	Specifies the time, in seconds, after the last modification before performing a rebase. Valid values are from 1 to 14400 seconds (4 hours).The default is 14400 seconds.
reset-period <i>value</i>	Specifies the period for performing a meta data refresh Valid values are from 1 to 900 seconds (15 minutes). The default is 900 seconds.

Command Modes Parameter map optimization configuration mode
Admin and user contexts

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines Rebasing refers to the process of updating the base file that is used for generating deltas between subsequent content retrievals. Because the base content of a site often changes over a period of time, the size of the generated deltas can grow relatively large. To maintain the effectiveness of the delta optimization process, the base files are automatically updated as required.

Examples To specify a rebase, based on a percentage of 1000 FlashForwarded URLs in the response, enter:

```
host1/Admin(config-parammap-optmz)# rebase flashforward-percent 1000
```

To revert to a default rebase setting, enter:

```
host1/Admin(config-parammap-optmz)# no rebase flashforward-percent
```

Related Commands This command has no related commands.

(config-parammap-optmz) request-grouping-string

To define a string to sort requests for AppScope reporting by the optional Cisco AVS 3180A Management Station, use the **request-grouping-string** command. Use the **no** form of the command to remove a request grouping string.

```
request-grouping-string string
```

```
no request-grouping-string string
```

Syntax Description	<i>string</i>	URL regular expression that defines a set of URLs. The string can contain the parameter expander functions listed in the (config-parammap-optmz) cache key-modifier section.
---------------------------	---------------	--

Command Modes Parameter map optimization configuration mode
Admin and user contexts

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines The string can contain a URL regular expression that defines a set of URLs in which URLs that differ only by their query parameters are to be treated as separate URLs in AppScope reports.

Typically, in an AppScope report organized by URL, matching URLs that differ only in their query parameters are treated as the same URL and are not listed on separate lines. Use the **request-grouping-string** command to specify that all URL variations that are based on query parameters are to be treated as separate URLs for reporting purposes. Each variation will appear on a separate line in the report.

For details about the Cisco AVS 3180A Management Station 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*.

Examples To define a string that is used to make the URLs `http://server/catalog.asp?region=asia` and `http://server/catalog.asp?region=america` into two separate reporting categories, enter:

```
host1/Admin(config-parammap-optmz) # request-grouping-string http_query_param(region)
```

To remove a request grouping string, enter:

```
host1/Admin(config-parammap-optmz) # no request-grouping-string
```

Related Commands [\(config-parammap-optmz\) appscope optimize-rate-percent](#)
[\(config-actlist-optm\) appscope](#)

(config-parammap-optmz) server-header

To define a user-specified string to be sent in the server header for an HTTP response, use the **server-header** command in parameter map optimization configuration mode. Use the **no** form of the command to delete the server header string.

server-header *string*

no server-header *string*

Syntax Description	
<i>string</i>	A particular string to be included in the server header. Enter a quoted text string. A maximum of 64 alphanumeric characters are allowed.

Command Modes Parameter map optimization configuration mode
Admin and user contexts

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines This command provide you with a method to uniquely tag the context or URL match statement by setting server header value to a particular string. The server header string can be used in cases where a particular URL is not being transmitted to the correct target context or the match statement.

Examples To specify a string to be sent in the server header, enter:

```
host1/Admin(config-parammap-optmz)# server-header "Header from Admin Context"
```

To delete the server header string, enter:

```
host1/Admin(config-parammap-optmz)# no server-header
```

Related Commands This command has no related commands.

(config-parammap-optmz) server-load

To control load-based expiration for the cache, use the **server-load** command. Use the **no** form of the command to revert to a default setting of 20 percent.

server-load { **trigger-percent** *value* | **ttl-change-percent** *value* }

no server-load { **trigger-percent** *value* | **ttl-change-percent** *value* }

Syntax Description	
trigger-percent <i>value</i>	Defines the threshold that triggers a change in the cache TTL. This keyword enables the ACE to monitor server load in real time and make intelligent “closed loop” content expiration decisions so that site performance is maximized and existing hardware resources are used most efficiently, even during periods of peak traffic load. Valid values are from 0 to 100 percent. The default is 20 percent.
ttl-change-percent <i>value</i>	Defines the percentage by which the cache TTL is increased or decreased in response to a change in the server load. For example, if you set this value to 20 and the current TTL for a particular response is 300 seconds, and if the current server response time exceeds the trigger threshold, then the cache TTL for the response is raised to 360 seconds (20 percent increase). Valid values are from 0 to 100 percent. The default is 20 percent.

Command Modes Parameter map optimization configuration mode
Admin and user contexts

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines Performance assurance with load-based expiration allows an object in the cache to expire (excluding the natural process of cache pruning). The origin server's load determines when the object expires.

This type of expiration allows you to dynamically increase the time to live (TTL) of cached responses if the current response time (average computed over a short time window) from the origin servers is larger than the average response time (average computed over a longer time window) by a threshold amount. Similarly, the TTL is dynamically decreased if the reverse holds true. The starting value for the cache TTL is the **cache ttl min** value (see the “[\(config-parammap-optmz\) cache ttl](#)” section) or 0 if you do not specify a value. Moving average-based calculation allows the cache to respond to trends in usage patterns, smoothing out uncharacteristic spikes.

Examples To specify a threshold trigger of 50 percent, enter:

```
host1/Admin(config-parammap-optmz) # server-load trigger-percent 50
```

To revert to a default setting of 20 percent, enter:

```
host1/Admin(config-parammap-optmz) # no server-load trigger-percent
```

Related Commands [\(config-parammap-optmz\) cache ttl](#)

(config-parammap-optmz) utf8 threshold

To determine how many UTF-8 characters on a page constitute a UTF-8 character set page for purposes of UTF-8 detection, use the **utf8 threshold** command. Use the **no** form of the command to disable the UTF-8 threshold.

utf8 threshold *value*

no utf8 threshold *value*

Syntax Description	<i>value</i>	Number of UTF-8 characters on a page that constitute a UTF-8 character set page. Valid values are from 1 to 1,000,000 characters. The default is 5 characters.
---------------------------	--------------	--

Command Modes	Parameter map optimization configuration mode Admin and user contexts
----------------------	--

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines	This threshold adjusts the detection of multibyte UTF-8 character set pages.
-------------------------	--

Examples	To specify a value of 1000 UTF-8 characters on a page, enter: host1/Admin(config-parammap-optmz)# utf8 threshold 1000
	To disable the UTF-8 threshold, enter: host1/Admin(config-parammap-optmz)# no utf8 threshold

Related Commands	This command has no usage guidelines.
-------------------------	---------------------------------------

Parameter Map SSL Configuration Mode Commands

Parameter map Secure Sockets Layer (SSL) configuration mode commands allow you to specify an SSL-type parameter map and configure SSL settings for the map. To create an SSL-type parameter map and access parameter map SSL configuration mode, use the **parameter-map type ssl** command in configuration mode. The prompt changes to (config-parammap-ssl). Use the **no** form of this command to remove the parameter map from the configuration.

parameter-map type ssl *name*

no parameter-map type ssl *name*

Syntax Description	<i>name</i>	Name assigned to the parameter map. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.
---------------------------	-------------	---

Command Modes	Configuration mode Admin and user contexts
----------------------	---

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines The commands in this mode require the connection or SSL feature in your user role. For details about role-based access control (RBAC) and user roles, see the *Cisco 4700 Series Application Control Engine Appliance Virtualization Configuration Guide*.

After you create and configure an SSL parameter map, you must associate the parameter map with a policy map to activate it. For details, see the [\(config-ssl-proxy\) ssl advanced-options](#) command in the “[SSL Proxy Configuration Mode Commands](#)” section.

Examples To create an SSL-type parameter map called SSL_MAP, enter:

```
host1/Admin(config)# parameter-map type ssl SSL_MAP
host1/Admin(config-parammap-ssl)#
```

Related Commands

- [\(config\) parameter-map type](#)
- [\(config-ssl-proxy\) ssl advanced-options](#)
- [show parameter-map](#)

(config-parammap-ssl) cipher

To define each of the cipher suites that you want the ACE to support during a secure session, use the **cipher** command. Use the **no** form of the command to delete a cipher suite from the SSL parameter map.

cipher *cipher_name* [**priority** *cipher_priority*]

no cipher *cipher_name*

Syntax Description		
<i>cipher_name</i>	Name of the cipher suite. See the “Usage Guidelines” section for the TCP options available for the available cipher suites that the ACE supports. Enter one of the supported cipher suites from Table 2-7 . The default setting is all .	
priority	(Optional) Assigns a priority level to the cipher suite. The priority level represents the preference-for-use ranking of the cipher suite, with 10 being the most preferred and 1 being the least preferred. By default, all configured cipher suites have a priority level of 1.	
<i>cipher_priority</i>	Priority level of the cipher suite. Enter a value from 1 to 10. The default priority value is 1.	

Command Modes	
	SSL parameter map configuration mode Admin and user contexts

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines [Table 2-7](#) lists the available cipher suites that the ACE supports and indicates which of the supported cipher suites are exportable from the ACE. [Table 2-7](#) also lists the authentication certificate and encryption key required by each cipher suite.

Table 2-7 Supported Cipher Suites

Cipher Suite	Exportable	Authentication Certificate Used	Key Exchange Algorithm Used
RSA_WITH_RC4_128_MD5	No	RSA certificate	RSA key exchange
RSA_WITH_RC4_128_SHA	No	RSA certificate	RSA key exchange
RSA_WITH_DES_CBC_SHA	No	RSA certificate	RSA key exchange
RSA_WITH_3DES_EDE_CBC_SHA	No	RSA certificate	RSA key exchange
RSA_EXPORT_WITH_RC4_40_MD5	Yes	RSA certificate	RSA key exchange
RSA_EXPORT_WITH_DES40_CBC_SHA	Yes	RSA certificate	RSA key exchange
RSA_EXPORT1024_WITH_RC4_56_MD5	Yes	RSA certificate	RSA key exchange

Table 2-7 Supported Cipher Suites (continued)

Cipher Suite	Exportable	Authentication Certificate Used	Key Exchange Algorithm Used
RSA_EXPORT1024_WITH_DES_CBC_SHA	Yes	RSA certificate	RSA key exchange
RSA_EXPORT1024_WITH_RC4_56_SHA	Yes	RSA certificate	RSA key exchange
RSA_WITH_AES_128_CBC_SHA	No	RSA certificate	RSA key exchange
RSA_WITH_AES_256_CBC_SHA	No	RSA certificate	RSA key exchange

Repeat the **cipher** command for each cipher suite that you want to include in the SSL parameter map.

The ACE chooses a cipher suite with the highest priority level from the client list. For SSL termination applications, the ACE uses the priority level to match cipher suites in the client's ClientHello handshake message. For SSL initiation applications, the priority level represents the order in which the ACE places the cipher suites in its ClientHello handshake message to the server.

The default "all cipher suites" setting works only when you do not configure the SSL parameter map with any specific ciphers. To return to using the "all cipher suites" setting, you must delete each of the specifically defined ciphers from the parameter map using the **no** form of the command.

Examples

To add the cipher suite RSA_WITH_AES_128_CBC_SHA and assign it a priority 2 level, enter:

```
host1/Admin(config-parammap-ssl)# cipher RSA_WITH_AES_128_CBC_SHA priority 2
```

To delete the cipher suite RSA_WITH_AES_128_CBC_SHA from the SSL parameter map, enter:

```
host1/Admin(config-parammap-ssl)# no cipher RSA_WITH_AES_128_CBC_SHA
```

Related Commands

[\(config-parammap-ssl\) version](#)

[show parameter-map](#)

(config-parammap-ssl) version

To specify the versions of Secure Sockets Layer (SSL) and Transport Layer Security (TLS) that the ACE supports when it uses the SSL proxy parameter map during the handshake process, use the **version** command. Use the **no** form of the command to remove a version from the SSL proxy parameter map.

```
version {all | ssl3 | tls1}
```

```
no version
```

Syntax Description	all	Specifies that the ACE supports both SSL (version SSL3) and TLS (version TLS1). This is the default setting.
	ssl3	Specifies that the ACE supports only SSL version SSL3.
	tls1	Specifies that the ACE supports only TLS version TLS1.

Command Modes	SSL parameter map configuration mode Admin and user contexts
---------------	---

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines	This command has no usage guidelines.
------------------	---------------------------------------

Examples	To specify the version SSL3, enter: <pre>host1/Admin(config-parammap-ssl)# version SSL3</pre> To remove the version TLS1 from the SSL proxy parameter map, enter: <pre>host1/Admin(config-parammap-ssl)# no version</pre>
----------	--

Related Commands	(config-parammap-ssl) cipher show parameter-map
------------------	--