

# Keepalive Configuration Mode Commands

Keepalive configuration mode allows you to configure keepalive properties and apply them to any service. Global keepalives reduce the amount of configuration required for each service. You can apply the keepalive configuration to multiple services. Global keepalives are independent of service mode.



## Note

You can configure keepalive message parameters for a service in service configuration mode. However, if you assign a keepalive you created in keepalive mode to a service, it supersedes the keepalive parameters configured in service mode.

The CSS divides the keepalive types into two categories, Class A and Class B keepalives. The CSS supports a maximum of 2048 Class A keepalives. The CSS supports a maximum of 512 Class B keepalives. Table 2-1 lists the keepalive types in each class, the maximum number of each type, and the maximum number of each keepalive type that can execute concurrently.

**Table 2-1** *Keepalive Class, Types, and Limitations*

Class	Type	CSS Maximum	Concurrent Maximum
A (The CSS limits 2048 keepalives per Class A.)	ICMP	2048	2048
	HTTP-HEAD non-persistent	2048	2048
	SSL (Hello)	2048	2048
	TCP	2048	2048
B (The CSS limits 512 keepalives per Class B.)	FTP	256	32
	HTTP-GET persistent and non-persistent	256	32
	HTTP-HEAD persistent	256	32
	Script	256	16

**Caution**

For an 11500 series CSS, do not configure more than 2048 total keepalives, including a total of 512 Class B keepalives. Any services assigned to keepalives over the supported total number will not be eligible for content rule selection.

Regardless of the number of services you assign to a global keepalive through the **(config-service) keepalive type named** command, the CSS always counts it as one keepalive.

To access keepalive configuration mode, use the **keepalive** command from circuit, global, header-field-group, interface, and IP configuration modes. The prompt changes to (config-keepalive [*name*]). You can also use this command from keepalive mode to access another keepalive. For information about commands available in this mode, see the following commands.

Use the **no** form of this command to delete an existing keepalive.

**keepalive** *name*

**no keepalive** *existing\_keepalive\_name*

**Syntax Description**

<i>name</i>	Name of a new keepalive you want to create or of an existing keepalive. Enter an unquoted text string with no spaces and a maximum length of 31 characters. To see a list of existing keepalive names, enter:
-------------	---

**keepalive ?**

**Related Commands**

**show keepalive**  
**(config-service) keepalive type named**

## (config-keepalive) active

To activate the keepalive you are configuring, use the **active** command. Activating a keepalive starts the sending of messages to the keepalive IP address.

**active**

---

### Related Commands

**show keepalive**  
**(config-keepalive) ip address**  
**(config-keepalive) suspend**

## (config-keepalive) description

To specify the description for the keepalive, use the **description** command. Use the **no** form of this command to delete the description.

**description** "*keepalive\_description*"

**no description**

---

### Syntax Description

*"keepalive\_description"* The description for the keepalive. Enter a quoted text string with a maximum length of 64 characters including spaces.

---

---

### Related Commands

**show keepalive**

## (config-keepalive) frequency

To specify the frequency to send keepalive messages to the IP address, use the **frequency** command. Use the **no** form of this command to reset the frequency to its default value of 5.

**frequency** *frequency*

**no frequency**

<b>Syntax Description</b>	<i>frequency</i>	Time in seconds between sending keepalive messages to the IP address. Enter an integer from 2 to 255. The default is 5.
---------------------------	------------------	---

**Usage Guidelines** For script keepalives, configure a higher frequency time value. A time interval of over 10 seconds ensures that the script keepalive has enough time to finish. Otherwise, state transitions may occur more often than usual.

If you configure more than 16 keepalives, the CSS automatically adjusts the keepalive frequency time to a value that best fits the resource usage. Note that this adjustment also affects the keepalive retry period value by adjusting that value to a number that is one-half the adjusted frequency time. If this occurs, you may observe in the running-configuration that your previously set keepalive frequency and retry-period times change to a different value, as determined by the CSS.

The timeout for a keepalive is related to the configured keepalive frequency. In WebNS 5.1 and earlier versions, the timeout is equivalent to the keepalive frequency. In version 5.2, the timeout is one second less than the keepalive frequency.

**Related Commands** `show keepalive`

## (config-keepalive) hash

To specify the MD5 hash for the keepalive, use the **hash** command. The keepalive process compares the hash with the computed hash of all HTTP GET responses. A successful comparison results in the keepalive maintaining an ALIVE state. Use the **no** form of this command to clear the hash value.

**hash** “*object*”

**no hash**

---

### Syntax Description

“*object*”

Object containing the MD5 hash in hexadecimal value for the keepalive. To determine the value for the hash, use the **show keepalive** command after you configure the keepalive without the hash option. Enter a quoted text string up to 32 characters.

---



---

### Related Commands

**show keepalive**

**(config-keepalive) http-rspcode** To specify the response code expected from the HTTP daemon when a CSS issues a HEAD request, use the **http-rspcode** command. This command could be helpful for checking a redirect by specifying the 302 response code, or by triggering another non-200 HTTP response code. Use the **no** form of the command to reset the response code to its default value of 200.

**http-rspcode** *value*

**no http-rspcode**

---

### Syntax Description

*value*

Response code expected from the HTTP daemon. Enter the response code as an integer from 100 to 999. The default is 200.

---



---

### Related Commands

**(config-service) keepalive http-rspcode**

## (config-keepalive) ip address

To specify the IP address where the keepalive messages are sent, use the **ip address** command.

**ip address** *ip\_address*

<b>Syntax Description</b>	<i>ip_address</i>	IP address for the keepalive. Enter this address in dotted-decimal notation (for example, 192.168.11.1).
---------------------------	-------------------	--

<b>Related Commands</b>	(config-keepalive) port
-------------------------	-------------------------

## (config-keepalive) maxfailure

To specify how many times the IP address can fail to respond to a keepalive message before being considered dead, use the **maxfailure** command. Use the **no** form of this command to reset the maximum failure number to its default value of 3.

**maxfailure** *number*

**no maxfailure**

---

<b>Syntax Description</b>	<i>number</i>	Maximum failure number. Enter an integer from 1 to 10. The default is 3.
---------------------------	---------------	--

---

## (config-keepalive) method

To specify the HTTP keepalive method assigned to the keepalive, use the **method** command.

**method** [*get|head*]

---

<b>Syntax Description</b>	<b>get</b>	Specifies the get method. The CSS issues a HTTP GET method to the service, computes a hash value on the page, and stores the hash value as a reference hash. Subsequent GETs require a 200 OK status (HTTP command completed OK response) and the hash value to equal the reference hash value. If the 200 OK status is not returned, or if the 200 OK status is returned but the hash value is different from the reference hash value, the CSS considers the service down.  When you specify the content information of an HTTP Uniform Resource Identifier (URI) for an HTTP keepalive, the CSS calculates a hash value for the content. If the content information changes, the hash value no longer matches the original hash value and the CSS assumes that the service is down. To prevent the CSS from assuming that a service is down due to a hash value mismatch, specify the <b>method</b> as <b>head</b> .
---------------------------	------------	---

---

---

**head** Specifies the head method (default). The CSS issues a HTTP HEAD method to the service and a 200 OK status is required. The CSS does not compute a reference hash value for this type of keepalive. If the 200 OK status is not returned, the CSS considers the service down.

---



---

### Usage Guidelines

If you change the keepalive method on an active service, suspend and reactivate the service for the change to take effect.

## (config-keepalive) no

To negate a command or set it to its default, use the **no** command. For information on general **no** commands you can use in this mode, see the general **no** command. The following options are available in keepalive mode.

---

### Syntax Description

<b>no acl</b> <i>index</i>	Deletes an existing ACL
<b>no description</b>	Clears the keepalive description
<b>no frequency</b>	Resets the keepalive frequency to its default of 5 seconds
<b>no hash</b>	Clears the MD5 hash object value
<b>no http-rspcode</b>	Resets the response code to its default value of 200
<b>no keepalive</b> <i>name</i>	Deletes an existing keepalive
<b>no maxfailure</b>	Resets the keepalive maximum number of failures to its default setting of 3
<b>no owner</b> <i>name</i>	Deletes an existing owner
<b>no port</b>	Resets the keepalive port number to its default setting based on the configured keepalive type
<b>no retryperiod</b>	Resets the keepalive retry period to its default of 5 seconds
<b>no uri</b>	Clears the keepalive content information of the URI

## (config-keepalive) port

To specify the port number for the keepalive, use the **port** command. Use the **no** form of this command to reset the port to the default based on the configured keepalive type.

**port** *number*

**no port**

---

**Syntax Description**

*number*

Port number associated with the keepalive. Enter the number as an integer from 0 to 65535. The default is based on the keepalive type. If the keepalive type is:

- Not configured, the default port number is 0
  - HTTP or TCP, the default port number is 80
  - FTP, the default port number is 21
- 

---

**Related Commands**

(config-keepalive) ip address

## (config-keepalive) retryperiod

To specify the retry period to send messages to the keepalive IP address, use the **retryperiod** command. Use the **no** form of this command to reset the retry period to its default value of 5.

**retryperiod** *period*

**no retryperiod**

<b>Syntax Description</b>	<i>period</i>	Time in seconds between sending retry messages to the keepalive IP address. Enter an integer from 2 to 255. The default is 5.
---------------------------	---------------	---

<b>Usage Guidelines</b>	When a service has failed to respond to a given keepalive message (the service has transitioned to the dying state), the retry period specifies how frequently the CSS tests the service to see if it is functional.
-------------------------	--

## (config-keepalive) suspend

To deactivate the keepalive, use the **suspend** command.

**suspend**

<b>Related Commands</b>	<b>show keepalive</b> <b>(config-keepalive) active</b>
-------------------------	---

## (config-keepalive) tcp-close

To specify the keepalive to close a TCP socket with a FIN or a RST, use the **tcp-close** command.

```
tcp-close [fin|rst]
```

### Syntax Description

<b>fin</b>	Specifies that the keepalive closes the TCP socket with a FIN rather than a RST.
<b>rst</b>	Specifies that the keepalive closes the TCP socket with a RST (default).

### Usage Guidelines

By default and in compliance with RFC 1122, the CSS sends a reset (RST) to close the socket on a server port for TCP keepalives. A RST is faster than a FIN, because a RST requires only one packet, while a FIN can take up to four packets. If your servers require a graceful closing of a socket using a FIN, use the **tcp-close fin** command.

The **tcp-close fin** and service mode **keepalive tcp-close fin** commands may be applied to a total of 100 TCP keepalives.

### Related Commands

(config-keepalive) type

## (config-keepalive) type

To specify the type of keepalive message assigned to the keepalive, use the **type** command.

```
type [ftp ftp_record|http {non-persistent}|icmp|script script_name
      {"arguments"} {use-output}|ssl|tcp]
```

<b>Syntax Description</b>	<b>ftp</b> <i>ftp_record</i>	The keepalive method by which the CSS logs in to an FTP server as defined in the FTP record file. Provide the name of an existing FTP record for the FTP server. Enter an unquoted text string with no spaces. To create an FTP record, use the <b>(config) ftp-record</b> command.
	<b>http</b> { <b>non-persistent</b> }	An HTTP index page request. By default, HTTP keepalives attempt to use persistent connections. To disable this behavior, include the <b>non-persistent</b> keyword.
	<b>icmp</b>	An ICMP echo message (default).
	<b>script</b> <i>script_name</i>	The script keepalive to be used by the service. The script is played every time the keepalive is entered. Enter the name of the script keepalive. To view a list of scripts, enter:  <b>type script ?</b>
	<i>“arguments”</i>	Arguments to pass into the keepalive script. Enter a quoted text string with a maximum of 128 characters including spaces.
	<b>use-output</b>	(Optional) Allows the script to parse the output for each executed command. This keyword allows the use <b>grep</b> and file direction within a script. By default, the script does not parse the output.  You can configure only 16 keepalives that use script output.
	<b>ssl</b>	SSL HELLO keepalives for this service. Use this keepalive for all backend services supporting SSL. The CSS sends a client HELLO to connect the SSL server. After the CSS receives a HELLO from the server, the CSS closes the connection with a TCP RST.  When the CSS is using an SSL module, use the keepalive type of <b>none</b> . The SSL module is an integrated device in the CSS and does not require the use of keepalive messages for the service.
	<b>tcp</b>	TCP connection handshake request.

**Usage Guidelines**

To enable the HTTP-HEAD optimization, use the **type http non-persistent** command.

The CSS divides the keepalive types into two categories, Class A and Class B keepalives. The CSS supports a maximum of 2048 Class A keepalives. The CSS supports a maximum of 512 Class B keepalives. Table 2-1 lists the keepalive types in each class, the maximum number of each type, and the maximum number of each keepalive type that can execute concurrently.

**Table 2-2 Keepalive Class, Types, and Limitations**

Class	Type	CSS Maximum	Concurrent Maximum
A  (The CSS limits 2048 keepalives per Class A.)	ICMP	2048	2048
	HTTP-HEAD non-persistent	2048	2048
	SSL (Hello)	2048	2048
	TCP	2048	2048
B  (The CSS limits 512 keepalives per Class B.)	FTP	256	32
	HTTP-GET persistent and non-persistent	256	32
	HTTP-HEAD persistent	256	32
	Script	256	16

**Caution**

For an 11500 series CSS, do not configure more than 2048 total keepalives, including a total of 512 Class B keepalives. Any services assigned to keepalives over the supported total number will not be eligible for content rule selection.

When the CSS is using an SSL module, use the keepalive type of **none**. The SSL module is an integrated device in the CSS and does not require the use of keepalive messages for the service.

The **tcp-close fin** and service mode **keepalive tcp-close fin** commands may be applied to a total of 100 TCP keepalives.

## (config-keepalive) uri

To specify the content information for an HTTP global keepalive, use the **uri** command. Use the **no** form of this command to clear the URI assigned to the keepalive.

**uri** *“uri”*

**no uri**

---

### Syntax Description

*“uri”*

Content information for the HTTP keepalive URI. Enter the content information for a URI as a quoted text string with a maximum length of 64 characters. Do not include the host information in the string. The CSS derives the host information from the service IP address and the keepalive port number.

---



---

### Usage Guidelines

When you specify the content information for an HTTP keepalive, the CSS calculates a hash value for the content. If the content information changes, the hash value no longer matches the original hash value and the CSS assumes that the service is down. To prevent the CSS from assuming that a service is down due to a hash value mismatch, specify the **keepalive method** as **head**. If you specify a Web page with changeable content and do not specify the keepalive method as **head**, you must suspend and reactivate the service each time the content information changes.