

# Service Configuration Mode Commands

Service configuration mode allows you to configure a service on the CSS. A service is an entity that contains and provides Internet content. It is identified by a name, an IP address, and optimally, a protocol and a port number. When you create a service, you can apply content rules to it. The rules allow the CSS to direct or deny requests for content from the service.

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

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

```
service service_name
no service service_name
```

---

## Syntax Description

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

---

## (config-service) access

To associate an access mechanism with a service for use during publishing, subscribing, and demand-based replication activities, use the **access** command. Use the **no** form of this command to remove a service access mechanism.

```
access ftp ftp_record
no access ftp
```

---

### Syntax Description

---

<i>ftp_record</i>	The name of an existing FTP record. Enter an unquoted text string with no spaces.
-------------------	---

---

---

### Usage Guidelines

You must use the **access** command for each service that offers publishing services. This command is optional for subscriber services; the subscriber service inherits the access mechanism from the publisher.

When you use this command to associate an FTP access mechanism to a service, the base directory of an existing FTP record becomes the tree root. To maintain coherent mapping between WWW daemons and FTP daemons, make the FTP access base directory equivalent to the WWW daemon root directory as seen by clients. For information on creating an FTP record, refer to the **(config) ftp-record** command.

---

### Related Commands

(config) **ftp-record**

## (config-service) active

To activate the specified service, use the **active** command. Activating a service puts it into the resource pool for load balancing content requests.

**active**

---

**Related Commands** (config-service) suspend

## (config-service) bypass-hosttag

To allow the Client Side Accelerator (CSA) on the CSS to bypass a cache farm and establish a connection with the origin server to retrieve non-cacheable content, use the **bypass-hosttag** command. The domain name from the host tag field is used to lookup the origin IP address on the CSA. Use the **no** form of this command to disable the bypassing of cache for non-cacheable content.

**bypass-hosttag**  
**no bypass-hosttag**

---

### Usage Guidelines

Before you can use this command, make sure that the service is suspended.

To bypass the cache farm for non-cacheable content, you must also configure a service IP address of 0.0.0.0 and a keepalive type of **none**.

---

**Related Commands** (config-service) ip address  
(config-service) keepalive type none  
(config-service) type

## (config-service) cache-bypass

To disable applying content rules to requests originating from a proxy or transparent-cache type service when the CSS processes the requests, use the **cache-bypass** command. By default, no content rules are applied to requests from a proxy or transparent-cache type service. Use the **no** form of this command to apply content rules to requests from a proxy or transparent-cache type service.

```
cache-bypass
no cache-bypass
```

---

**Related Commands** (config-service) type

## (config-service) cookie

To specify the HTTP cookie for the service, use the **cookie** command. This command is replaced by the **(config-service) string** command.

```
cookie cookie_name
```

---

<b>Syntax Description</b>	<i>cookie_name</i>	The name of the cookie. Enter a unquoted text string with no spaces and a maximum length of 15 characters.
---------------------------	--------------------	--

---

## (config-service) domain

To specify the domain name to prepend to a requested piece of content when an HTTP redirect service generates an “object moved” message for the service, use the **domain** command. Use the **no** form of this command to clear the redirect domain for the service.

**domain** *domain\_name*  
**no domain**

### Syntax Description

<i>domain_name</i>	The name of the domain. Enter a unquoted text string with no spaces and a maximum length of 64 characters.
--------------------	--



**Note** The CSS automatically prepends the domain name with http://.

### Usage Guidelines

The CSS uses the configured domain name in the redirect message as the new location for the requested content. The CSS prepends the domain name to the requested URL. If the domain name is not configured, the CSS uses the domain in the host-tag field from the original request. If no host tag is found, the CSS uses the IP address of the service to generate the redirect.



**Note** You can only use a service redirect domain on a service of type redirect.



**Note** The **redirect-string** and **(config-service) domain** commands are similar. The CSS returns the **redirect-string** command string verbatim as configured. With the **(config-service) domain** command, the CSS prepends the domain to the original requested URL.

**Note**

You cannot simultaneously configure the **domain** and **(config-service) redirect-string** commands on the same service.

**Related Commands**    `show service`

## (config-service) ip address

To specify the service IP address or a range of addresses, use the **ip address** command. Use the **no** form of this command to clear the IP address for a service and set it to its default value of 0.0.0.0.

```
ip address ip_address {range number}
no ip address
```

**Syntax Description**

<i>ip_address</i>	The IP address for the service. Enter the address in dotted-decimal notation (for example, 192.168.11.1). The default is 0.0.0.0.
<b>range</b> <i>number</i>	The <b>range</b> option allows you to specify a range of IP addresses starting with the IP address ( <i>ip_address</i> ). Enter a number from 1 to 65535. The default range is 1.  For example, if you enter an IP address of 203.1.1.1 with a range of 10, the IP addresses range from 203.1.1.1 through 203.1.1.10.

**Usage Guidelines**

Before you can change the address, make sure that the service is suspended.

**Related Commands**

**(config-service) port**

## (config-service) keepalive

To configure keepalive message parameters for the service, use the **keepalive** command. The options for this service mode command are:

- **keepalive frequency**..., specifies the keepalive message frequency
- **keepalive hash**..., specifies the MD5 hash for the keepalive
- **keepalive maxfailure**..., specifies how many times the service can fail to respond to a keepalive message before it is considered offline
- **keepalive method**..., specifies the HTTP method for the service
- **keepalive port**..., specifies the keepalive port
- **keepalive retryperiod**..., specifies the keepalive retry period for the service
- **keepalive type**..., specifies the type of keepalive message, if any, appropriate for the service
- **keepalive uri**..., specifies the HTTP keepalive URI for the service

For more information on these options and associated variables, refer to the following commands.

### Usage Guidelines

With keepalive messages, you can determine whether a service is still functioning. The CSS periodically sends a keepalive message to a service to determine the state of the service. The CSS considers the service to be alive when a service responds to the keepalive message.

The CSS transitions the service to the dying state when the service fails to respond to a keepalive message. The CSS tests whether the failed service is functional by sending a keepalive message at time intervals based on the retry period.

The CSS transitions the service to the dead state if the service fails to respond a maximum number of retries to the keepalive message. Then the CSS removes the service from the load-balancing algorithm. The CSS continues to test whether the service is functional at time intervals based on the retry period.

Thus, using the default values of a 5-second keepalive frequency interval, a 5-second retry period interval, and maximum of three failures, a service can transition from the alive state to the dead state in 20 seconds; a 5-second interval

between a keepalive response and the initial keepalive failure based on the keepalive frequency, and three failures, each occurring at 5-second intervals based on the retry period.

The CSS supports a maximum of 512 keepalives (255 script keepalive types). These keepalives include:

- ICMP, HTTP, FTP, and TCP keepalives configured and assigned to a service through this service mode command. Each time you assign one of these keepalives to a service through the **(config-service) keepalive type** command, the CSS counts it as one keepalive.
- Global keepalives configured through the **(config) keepalive** command. 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.



#### Caution

---

Do not configure more than 512 individual keepalives. Any services assigned to keepalives over 512 will not be eligible for content rule selection.

---

## keepalive frequency

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

**keepalive frequency** *frequency*  
**no keepalive frequency**

---

#### Syntax Description

<i>frequency</i>	The time in seconds between sending keepalive messages to the service. 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.

---

**Command Modes** Service

## keepalive hash

To specify the MD5 hash for the keepalive, use the **keepalive 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.

**keepalive hash** "*object*"  
**no keepalive hash**

---

<b>Syntax Description</b>	<i>object</i>	The object containing the MD5 hash in hexadecimal value for the keepalive. To determine the value for the hash, use the <b>show keepalive</b> command after you configure the keepalive without the hash option. Enter a quoted text string up to 32 characters.
---------------------------	---------------	--

---



---

**Command Modes** Service

---

**Related Commands** **show keepalive**

---

**Command Modes** Service

## keepalive maxfailure

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

**keepalive maxfailure** *number*  
**no keepalive maxfailure**

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

<b>Command Modes</b>	Service
----------------------	---------

## keepalive method

To specify the HTTP keepalive method for the service, use the **keepalive method** command.

**keepalive method** [gethead]

<b>Syntax Description</b>	<b>get</b>	<p>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.</p> <p>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>keepalive method</b> as <b>head</b>.</p>
	<b>head</b>	<p>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.</p>

<b>Command Modes</b>	Service
----------------------	---------

## keepalive port

To define a port number for the keepalive, use the **keepalive port** command. Use the **no** form of this command to reset the keepalive port to its default setting.

**keepalive port** *number*  
**no keepalive port**

---

### Syntax Description

*number* The port number for the keepalive. Enter the number as an integer from 0 to 65535. The default setting is based on the configured service port number. Otherwise, the default setting 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

---

---

### Command Modes

Service

---

### Usage Guidelines

If you do not configure the port, the keepalive uses the service port configured with the **(config-service) port** command. If you do not configure either port, the keepalive is based on the configured keepalive type.

---

### Related Commands

**(config-service) keepalive type**

## keepalive retryperiod

To specify the keepalive retry period for the service, use the **keepalive retryperiod** command. Use the **no** form of this command to reset the retry period to its default value of 5.

**keepalive retryperiod** *period*  
**no keepalive retryperiod**

---

### Syntax Description

<i>period</i>	The time in seconds between sending retry messages to the service. Enter an integer from 2 to 255. The default is 5.
---------------	--

---



---

### Command Modes

Service

---


### Usage Guidelines


When a service has failed to respond to a given keepalive message (the service is now transitioned to the dying state), the retry period specifies how frequently the CSS tests the service to see if it is functional.

## keepalive type

To specify the type of keepalive message, if any, appropriate for the service, use the **keepalive type** command.

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

Syntax Description		
<b>ftp</b>		The keepalive method that accesses an FTP server by logging into an FTP server as defined in an FTP record file.
<i>ftp_record</i>		The name of an existing FTP record for an 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> option.
<b>icmp</b>		An ICMP echo message (default).
<b>named</b> <i>name</i>		A global keepalive defined in keepalive configuration mode. To view a list of defined keepalive names, enter:  <pre><b>keepalive type named ?</b></pre> Before using this command, make sure that the keepalive is activated through the <b>(config-service) active</b> command.
		
	<b>Note</b>	Assigning this global keepalive to a service overrides any keepalive properties configured in service mode.
<b>none</b>		Do not send keepalive messages to the service.

<b>script</b>	The script keepalive to be used by the service. The script is played every time the keepalive is issued.
	 <b>Note</b> You can configure only 255 script keepalives.
<i>script_name</i>	The name of the script keepalive. To view a list of scripts, enter: <pre>type script ?</pre>
<i>arguments</i>	The optional arguments to pass into the keepalive script. Enter a quoted text string with a maximum of 128 characters including spaces.
<b>use-output</b>	Allows the script to parse the output for each executed command. This optional keyword allows the use <b>grep</b> and file direction within a script. By default, the script does not parse the output.
	 <b>Note</b> You can configure only 16 keepalives that use script output.
<b>tcp</b>	The TCP connection handshake request. To define a port for a TCP keepalive, use the ( <b>config-service</b> ) <b>keepalive port</b> command.

**Usage Guidelines**

You can configure a maximum of 512 keepalive per type except script keepalives. You can configure only 255 script keepalives. Note that the CSS support a maximum of 512 at a time.

**Command Modes**

Service

## keepalive uri

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

**keepalive uri** *“uri”*

**no keepalive uri**

---

### Syntax Description

*uri*

The HTTP keepalive URI for the service. Enter the the content information of the URI as a quoted text string with a maximum 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 of a 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, define **keepalive method** as **head**. The CSS does not compute a hash value for this type of keepalive.

If you specify a Web page with changeable content and do not specify the head keepalive method, you must suspend and reactivate the service each time the content changes.

---

### Command Modes

Service

## (config-service) max age

To define the maximum age for replicated objects on services defined as type **rep-cache-redis**, **rep-store**, or **rep-store-redis**, use the **max age** command. The CSS deletes the dynamic content rule after the maximum age time elapses. Use the **no** form of this command to set the maximum age for replicated objects to its default value of 120.

```
max age minutes
no max age
```

<b>Syntax Description</b>	<i>minutes</i>	The maximum time in minutes. Enter a number from 1 to 1440. The default value is 120.
---------------------------	----------------	---

## (config-service) max connections

To define the maximum number of TCP connections on the services, use the **max connections** command. Use the **no** form of this command to set the maximum TCP connections to the default of 0.

```
max connections number
no max connections
```

<b>Syntax Description</b>	<i>number</i>	The maximum number of TCP connections on the service. Enter a number from 0 to 65535. The default is 0.
---------------------------	---------------	---

<b>Usage Guidelines</b>	Do not use service max connections on UDP content rules. The service connection counters do not increment and remain at 0 because UDP is a connectionless protocol.
-------------------------	---

## (config-service) max content

To define the maximum pieces of content for replication on services defined as type **rep-cache-redir**, **rep-store**, or **rep-store-redir**, use the **max content** command. Use the **no** form of this command to set the maximum content to its default value of 100.

**max content** *number*  
**no max content**

---

**Syntax Description**

---

<i>number</i>	The maximum content for replication. Enter a number from 1 to 65535. The default is 100.
---------------	--

---

## (config-service) max usage

To define the maximum disk space allowed for replication on services defined as type **rep-cache-redir**, **rep-store**, or **rep-store-redir**, use the **max usage** command. Use the **no** form of this command to set the maximum disk space to its default value of 1 megabyte.

**max usage** *mbytes*  
**no max usage**

---

**Syntax Description**

---

<i>mbytes</i>	The maximum disk space in megabytes. Enter a number from 1 to 1000. The default is 1.
---------------	---

---

## (config-service) 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, refer to the general **no** command. The following option is available in service mode.

Syntax Description		
<b>no access ftp</b>		Removes the service access mechanism.
<b>no acl <i>index</i></b>		Deletes an ACL.
<b>no bypass-hosttag</b>		Disables the bypassing of cache for non-cacheable content.
<b>no cache-bypass</b>		Allows the applying of content rules to requests from a proxy or transparent cache service.
<b>no domain</b>		Clears the redirect domain for the service
<b>no ip address</b>		Clears the IP address for the service and set it to its default value of 0.0.0.0.
<b>no keepalive frequency</b>		Resets the keepalive frequency to its default value of 5 seconds.
<b>no keepalive hash</b>		Clears the keepalive MD5 hash object.
<b>no keepalive maxfailure</b>		Resets the keepalive maximum failures to its default value of 3.
<b>no keepalive port</b>		Resets the keepalive port to its default setting based on configured service port. Otherwise, the default setting is based on the configured keepalive type.
<b>no keepalive retryperiod</b>		Resets the keepalive retry period to its default value of 5 seconds.
<b>no keepalive uri</b>		Clears the content information for the HTTP keepalive URI.
<b>no max age</b>		Resets the maximum age for replicated content to the default of 120 minutes.
<b>no max connection</b>		Resets the maximum TCP connections on the service to the default of 0.
<b>no max content</b>		Resets the maximum content for replication to the default of 100 pieces.

<b>no max usage</b>	Resets the maximum disk space allowed for replication to the default of 1 megabyte.
<b>no owner</b> <i>existing_owner_name</i>	Deletes an existing owner.
<b>no port</b>	Resets the IP port for the service to the default of <b>any</b> .
<b>no protocol</b>	Resets the IP protocol for the service to the default of <b>any</b> .
<b>no publisher</b>	Removes publishing on a service.
<b>no publisher interval</b>	Disables the publisher resynchronization interval by setting it to its default of 0.
<b>no redirect-string</b>	Removes the redirect string from the service.
<b>no rmon-event</b> <i>index</i>	Deletes an RMON event.
<b>no rmon-history</b> <i>index</i>	Deletes an RMON history.
<b>no string</b>	Removes the cookie from the service.
<b>no subscriber</b>	Unsubscribes the service from a publishing service.
<b>no transparent-hosttag</b>	Disables destination NATing for the transparent cache service type.
<b>no type</b>	Resets the type for the service to its default setting of local.
<b>no weight</b>	Resets the service weight to its default setting of 1.

## (config-service) port

To specify the service TCP/UDP port number or a range of port numbers, use the **port** command. Use the **no** form of this command to reset the port to **any**.

```
port number1 {range number2}
no port
```

### Syntax Description

<i>number1</i>	The TCP or UDP destination port number associated with a service. Enter the number from 0 to 65535. The default is <b>any</b> .
<b>range</b> <i>number2</i>	The <b>range</b> option allows you to specify a range of ports starting with the port <i>number1</i> . Enter a number from 1 to 65535. The default range is 1.  For example, if you enter a port number of 101 with a range of 10, the ports range from 101 through 110.

### Usage Guidelines

Before you can change the port, make sure that the service is suspended.

### Related Commands

(config-service) ip address  
(config-service) protocol

## (config-service) protocol

To specify the service IP protocol, use the **protocol** command. The default setting for this command is **any**, for any IP protocol. Use the **no** form of this command to reset the protocol to the default of **any**.

```
protocol [tcp|udp]
no protocol
```

---

### Syntax Description

---

<b>tcp</b>	The service uses the TCP protocol suite.
<b>udp</b>	The service uses the UDP protocol suite.

---

---

### Usage Guidelines

Before you can change the protocol, make sure that the service is suspended.

---

### Related Commands

(config-service) ip address  
(config-service) keepalive type  
(config-service) port

## (config-service) publisher

To configure a service as a publishing service and define its synchronization interval, use the **publisher** command. Use the **no** form of this command to remove publishing on a service or disable the publisher resynchronization interval by setting it to its default of 0.

```
publisher {interval minutes {trigger_file}}
no publisher
no publisher interval
```

### Syntax Description

<b>interval</b>	Defines a recurrent interval in minutes to synchronize content among the subscribers. You can only issue this command after you configure the service as a publishing service.
<i>number</i>	The synchronization interval in minutes. Enter the number from 0 to 3600. The default is 0 which disables the interval.
<i>trigger_file</i>	Optional path and filename to a file, when modified, triggers the publishing service to synchronize the content among the subscribers. Enter an unquoted character string with a maximum of 64 characters.

### Usage Guidelines

Use the **publisher** command to configure a service as a publishing service.

A publishing service can synchronize content among associated subscriber services. To move the content during publishing activities, configure an access mechanism by using the **(config-service) access** command.

When you define the interval to synchronize the subscriber, the interval begins at the time that you issue the command. Subscribers that are unavailable for synchronization are placed in an offline state and retried until the operation is completed.

**Note**


---

The publisher service does not become active until it has at least one configured subscriber. You do not need to configure the publisher before configuring the subscriber; but the publisher must be configured before the subscriber can receive any content synchronization updates.

---

**Related Commands**

**replicate**  
**(config) ftp-record**  
**(config-service) access**  
**(config-service) subscriber**

## **(config-service) redirect-string**

To specify an HTTP redirect string to be used when an HTTP redirect service generates an “object moved” message for the service, use the **redirect-string** command. Use the **no** form of this command to remove the redirect string from the service.

**redirect-string** *string*  
**no redirect-string**

**Syntax Description**


---

<i>string</i>	The HTTP redirect string. Enter a unquoted text string with no spaces and a maximum length of 64 characters.
---------------	--

---

**Usage Guidelines**

The CSS uses the entire configured redirect string as the new location for the requested content. If no string is configured, the CSS prepends the domain configured with the **(config-service) domain** command to the original request. If neither the redirect string or domain name are configured, the CSS uses the domain in the host-tag field from the original request combined with the requested HTTP content URL. If no host tag is found, the CSS uses the IP address of the service to generate the redirect.



---

**Note** You can only use a redirect string on a service of type redirect.

---



---

**Note** The **redirect-string** and **(config-service) domain** commands are similar. The CSS returns the **redirect-string** command string verbatim as configured. However, the CSS prepends the domain configured with the **(config-service) domain** command to the original requested URL.

---



---

**Note** You cannot simultaneously configure the **redirect-string** and **(config-service) domain** commands on the same service.

---

## (config-service) string

To specify the HTTP cookie for the service, use the **string** command. Use the **no** form of this command to remove the cookie for the service.

```
string cookie_name  
no string
```

---

**Syntax Description**

---

<i>cookie_name</i>	The name of the cookie. Enter a unquoted text string with no spaces and a maximum length of 15 characters.
--------------------	--

---

## (config-service) subscriber

To configure a service as a subscriber to a publishing service, use the **subscriber** command. Use the **no** form of this command to unsubscribe the service from a publishing service.

```
subscriber publisher  
no subscriber
```

---

**Syntax Description**

---

<i>publisher</i>	The name of the publishing service
------------------	------------------------------------

---

---

**Usage Guidelines**

By default, the subscriber inherits the access mechanism of the publisher for the movement of content. But if you want to configure an alternative mechanism, use the **(config-service) access** command.

---

**Related Commands**

(config) ftp-record  
(config-service) access  
(config-service) publisher

## (config-service) suspend

To remove the service from the pool for future load balancing content requests, use the **suspend** command. Suspending a service does not affect existing content flows, but it does prevent additional connections from accessing the service for its content.

### suspend

**Note**

If you suspend a service, the CSS uses the **failover** command setting to handle content requests.

---

**Related Commands** (config-service) active

## (config-service) transparent-hosttag

To enable destination network address translation (NAT) for the transparent cache service type, use the **transparent-hosttag** command. Use the **no** form of this command to disable destination network address translation for the transparent cache service type.

**transparent-hosttag**  
**no transparent-hosttag**

---

**Usage Guidelines**

Before you can use this command, make sure that the service is suspended.

Currently, you can use this command only in a CSA environment.

You do not need to configure source groups in a CSA environment. The transparent cache environment does not require the client source IP NATing that occurs as a result of a source group configuration.

---

**Related Commands** (config-service) type

## (config-service) type

To specify the type for the service, use the **type** command. If you do not define a type for the service, the default service type is local. Use the **no** form of this command to reset the type for the service to its default setting of local.

```
type [nci-direct-return|nci-info-only|proxy-cache|redirect
|redundancy-up|rep-cache-redir|rep-store|rep-store-redir
|transparent-cache]
no type
```

Syntax Description	
<b>nci-direct-return</b>	Specifies a NAT Channel Indication (NCI) service for NAT peering. NAT peering allows the building of forward TCP switched connections between CSSs until the destination CSS is reached and the destination CSS performs the final transformations, which allows return traffic packets to flow to the client through any network path. This service type informs the CSS to include the NCI option in the TCP packet. This option indicates to the server-side CSS that NAT parameters are in use, and contains the original source and destination IP addresses, and TCP port numbers. If a Layer 5 rule is matched, the spoof bit in the NCI option is set to indicate that part of the flow has been spoofed and the rest of the forward path must be established before the destination CSS can use the information in the packet to perform the NAT transformations for the reverse path. Configure the VIP for the service to the VIP on the server-side CSS to indicate an endpoint for the connection.
<b>nci-info-only</b>	Specifies the service is NAT Channel indication for information only.

<b>proxy-cache</b>	<p>Specifies the service is a proxy cache. This option bypasses content rules for requests from the cache. Bypassing content rules prevents a loop from forming between the cache server and the CSS. To allow the applying of content rules to requests, enter:</p> <p style="text-align: center;"><b>no cache-bypass</b></p>
<b>redirect</b>	<p>Specifies the service is not directly accessible and requires redirection. The CSS must use the HTTP redirect mechanism to direct the client request to the desired content.</p>
<b>redundancy-up</b>	<p>Specifies the service is a redundant uplink. You cannot use this service type and the <b>(config) ip redundancy master</b> command simultaneously. Before you can specify a redundant uplink, you must issue the <b>(config) no ip redundancy master</b> command.</p>
<b>rep-cache-redir</b>	<p>Specifies the service is a replication cache with redirect. The CSS uses the replication cache as a redirect service instead of load balancing between the local service and the cache.</p>
<b>rep-store</b>	<p>Specifies the service is a replication store server for hot content. The service is a local overflow service used to load balance content requests. The CSS moves hot content to the server. Then, the CSS creates a dynamic content rule for the hot content automatically. The dynamic content rule inherits all the attributes of the existing rule with the following changes:</p> <ul style="list-style-type: none"> <li>• Specifically identifies the hot content</li> <li>• Changes the server type from replication-store to type local</li> </ul> <p>The CSS deletes the dynamic content rule after the maximum age time elapses or the service keepalive indicates failure.</p>

---

<b>rep-store-redirect</b>	Specifies the service is a replication store to which content requests are redirected. The service is a remote overflow service. No content rules are applied to requests from this service type.
<b>transparent-cache</b>	Specifies the service is a transparent cache. No content rules are applied to requests from the cache. Bypassing content rules prevents a loop from forming between the cache server and the CSS. To allow the applying of content rules to requests, enter:  <b>no cache-bypass</b>

---

---

**Usage Guidelines**

Before you can change the type, make sure that the service is suspended.

## (config-service) weight

To specify the relative weight of the service, use the **weight** command. The weight is used in ArrowPoint Content Awareness (ACA) and weighted round-robin load balancing decisions. Use the **no** form of this command to reset the service weight to its default value of 1.

**weight** *weight*  
**no weight**

---

### Syntax Description

---

*weight* The service weight used with load metrics to make load allocation decisions. You can use the weight to bias flows toward the specified service. Enter an integer from 1 to 10. The default is 1.

---



---

### Usage Guidelines

The weight for the service set through the **(config-owner-content) add service** command takes precedent over the **(config-service) weight** command.

---

### Related Commands

**(config-owner-content) add service**  
**(config-owner-content) balance**