

# Configuring Persistence, Remapping, and Redirection on the CSS 11000 and CSS 11500

Document ID: 12646

## Contents

### Introduction

#### Before You Begin

Conventions

Prerequisites

Components Used

#### Content Rule Persistence

#### Bypass Persistence

#### HTTP Redirection and Service Remapping

#### Issuing the show remap Command

#### Related Information

## Introduction

During the life of a persistent connection, the Cisco Content Services Switch (CSS) determines when to move a client connection to a new service based on content rules, load balancing, and service availability. Some situations do not require moving the client connection; in others, it is mandatory. This document describes how to configure the CSS to make these decisions using the following:

- Content Rule Persistence
- Bypass Persistence
- HTTP Redirection and Service Remapping

## Before You Begin

### Conventions

For more information on document conventions, see the Cisco Technical Tips Conventions.

### Prerequisites

There are no specific prerequisites for this document.

### Components Used

This document is not restricted to specific software and hardware versions.

The information presented in this document was created from devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If you are working in a live network, ensure that you understand the potential impact of any command before using it.

## Content Rule Persistence

When a CSS receives a request for content from a client, the software compares the request to a content rule and determines which service handles the request. If the request matches a content rule, the CSS establishes a client connection to the service specified by the content rule. By default, the CSS keeps the client on the same connection for an entire flow session as long as a new content request meets the following conditions:

- Matches the same content rule that specified the current service.
- Matches a new content rule that contains the current service, even if a different service is specified by the content rule.
- Does not match a content rule, but a previous content rule match connected the client to the current service.

This CSS behavior is known as content rule persistence. When using transparent caches, which pre-fetch content, or mirrored content servers, this scheme works because the same content is available on each service.

The **persistent** command in content configuration mode maintains a persistent connection with a server as long as the above criteria is met. Persistence is enabled by default. Disabling persistence allows the CSS to move a connection to a better service on the same rule or utilize cache bypass functionality, Extension Qualifier Lists (EQLs), or failover bypass. The **no persistent** command should be configured on a content rule with the following:

- A balance method of domain or domain hash when using proxy caches.
- A method of url or urlhash when using transparent caches.
- A failover method of bypass when using transparent caches.
- An EQL bypass using transparent cache.
- Adding a sorry server to a content rule.

Issue the following command to enable persistence:

```
(config-owner-content)# persistent
```

Issue the following command to disable persistence:

```
config-owner-content)# no persistent
```

The CSS redirects a request for content on a persistent connection if it matches a new content rule that does not contain the current service, or if persistence is disabled and there is a better service configured in the rule. The CSS redirects or remaps the current connection to a new service based on the **persistence reset** command setting if the command is configured. If the **persistence reset** command is not configured, the CSS performs a HTTP redirect by default.

## Bypass Persistence

If a CSS bypasses a service (for example, with transparent cache down and failover bypass configured) when the next content request on the same TCP connection matches a content rule that contains the down transparent cache, the CSS will continue to bypass the cache, by default, even after the bypassed service is back online. In this case, the CSS typically sends the content request to the origin server. This behavior is called bypass persistence.

Configure the CSS to redirect or remap a bypassed connection using the **bypass persistence** global configuration command in conjunction with the **persistence reset** command.

Issue the **bypass persistence** command to determine if the CSS redirects or remaps operation properly. The CSS should redirect to reset a bypassed service when a content request matches a content rule and a previous request caused the bypass. This global command affects all operation flows.

Bypass persistence is enabled by default with the following command:

```
(config)# bypass persistence disable
```

CSS redirects or remaps to reset the connection according to the setting of the persistence reset method.

```
(config)# bypass persistence enable
```

CSS does not redirect or remap to reset the connection and continues to bypass a service.

## HTTP Redirection and Service Remapping

Content rule persistence is not useful when placing different content on different servers (for example, to conserve server disk space for load balancing considerations, or when using proxy caches). Disable persistence by issuing the **no persistent** command in these situations.

When the CSS receives a client request for content not available on the current service, reset the connection to the current service and establish a new connection to the service containing the requested content, (for example, a different proxy cache or the origin server). This is accomplished by one of the following ways:

- **Redirection:** An HTTP technique that resets the client-to-CSS (front end) connection and the CSS-to-service (back end) connection, then establishes a new flow to the best service containing the requested content.
- **Service Remapping:** A technique that resets the back end connection and moves the back end connection to the best service containing the requested content. This technique is faster and more efficient than redirection since the CSS does not reset and reestablish the front end connection. With service remapping, the CSS strictly manages port mapping to prevent the occurrence of duplicate port numbers.

Service remapping is used when moving the back end connection satisfies the content request and rule match. The CSS uses redirect to move the client connection to a new VIP address and/or port.

When resetting a connection to a new back end service, issue the **persistence reset** command with the **no persistent** command to cause an HTTP redirection or perform a back end remapping operation. This global command affects all flow setups that require redirecting or remapping.

Issue the following command to enable redirection:

```
(config)# persistence reset redirect
```

Issue the following command to enable service remapping:

```
(config)# persistence reset remap
```

The CSS does not use remapping when selecting redirect type services.

## Issuing the show remap Command

Issue the **show remap** command to display the configured persistence reset and bypass persistence settings and to view the megamaps used. This command is available in all modes.

```
# show remap
Persistence Reset Method: Redirect
Bypass Persistence: Disabled
```

## Related Information

- [Cisco CSS 11000 Series Product Support Page](#)
  - [Technical Support – Cisco Systems](#)
- 

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2014 – 2015 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

---

Updated: Dec 27, 2007

Document ID: 12646

---