

# Table of Contents

<b><u>Configuring ACA on the CSS 11000 and 11500</u></b> .....	<b>1</b>
<u>Document ID: 12579</u> .....	1
<u>Introduction</u> .....	1
<u>Prerequisites</u> .....	1
<u>Requirements</u> .....	1
<u>Components Used</u> .....	1
<u>Conventions</u> .....	1
<u>Information</u> .....	1
<u>Command Summary</u> .....	2
<u>Related Information</u> .....	3

# Configuring ACA on the CSS 11000 and 11500

Document ID: 12579

---

**Introduction**

**Prerequisites**

Requirements

Components Used

Conventions

**Information**

**Command Summary**

**Related Information**

---

## Introduction

This document provides information on how to configure ArrowPoint Content Aware (ACA) on the Content Services Switch (CSS) 11000 and 11500.

**Note:** CSS 11000 is now End-of-Life. For more information, refer to End-of-Sale Models and bulletins.

## Prerequisites

### Requirements

There are no specific requirements for this document.

### Components Used

The information in this document is based on these software and hardware versions:

- All CSS 11000 and 11500 Content Smart Web Switches
- CiscoWebNS software releases

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

### Conventions

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

## Information

The ACA algorithm is a load balancing algorithm that was designed to optimize a web site s performance because it prefers servers with better response times over servers with worse response times. The Content Smart Web Switch (such as the CSS 11000 and 11500) uses the normalized average response times from synchronize/start (SYN) to finish (FIN) of the TCP connections between clients and servers to determine a load value for each server. The server with the lowest average response time is given a load value of 2. The load ranges are from 2 to 255, and they are calculated separately for long and short (<15K) files and then

averaged together. A server with a value of 255 is brought out of rotation. As the load value increases for a server, the less requests ACA sends to it. The algorithm that ACA uses to pull the server out of rotation is not linear, therefore, the server with a load value of 200 is hit much, much less than a server with a load value of 12, as compared with a server with a load of 2.

There are two parameters that can be used to modify the load values of the servers. The first is the load step. The load step is 10 milliseconds by default. The load step refers to the difference between each load value. A service with a load of 3 compared to a service with a load of 2 has a response time that is 10 milliseconds worse. A service with a load of 10 compared to a service with a load of 2 has a response time that is 80 milliseconds worse.

The second value that can be used to modify load is the load teardown-timer. The load teardown-timer is used to set the maximum time between teardown reports. Teardown reports are used to derive the load number for a service; they include a summary of response times for services when flows are being torn down. If your CSS does not have a high volume of flows, you may want to decrease the load teardown-timer so that the load value gets calculated more often, and therefore causes the CSS to become more reactionary when you use ACA.

There are two values that can be adjusted to modify the behavior of ACA. The load-threshold is used to determine when a service should be brought out of rotation. The default value is 254. If you determine that your site's performance is hindered when a service reaches a load of 100, you can change the load threshold to 99, which would cause these servers to come out of rotation. Since ACA uses the load value to determine how many requests it should send a server, past performance values are used to make load balancing decisions. If a service is not hit, the CSS has no data to make future load balancing decisions. Periodically, ACA causes the "bad" service to be hit so that it can determine if it has recovered. The load ageout-timer is the time in seconds that the CSS does not hit a server that has exceeded the load threshold. If you want to attempt to bring the server back sooner, you could decrease this value; the default is 60 seconds. If your server typically takes a longer time to recover than 60 seconds, you want to increase this value.

## Command Summary

Issue these commands on the CSS:

- **show load** this command provides the average of long and short loads.
- **load step msec dynamic** this command is used to decrease to create greater deltas in load, or to increase to create smaller deltas in load.

```
load step 20 static
!--- Services with response times within 20 milliseconds
!--- will be treated the same.
```

- **load teardown-timer seconds** this command is used to decrease, which makes the load more reactionary (low flow rate).

```
load teardown-timer 10
!--- Decreased for low flow-rate.
```

- **load ageout-timer seconds** this command is used to increase to take longer to bring back in the "bad" server or decrease, to bring the "bad" server back more quickly.

```
load ageout-timer 120
```

*!--- Increased for slow recovering server.*

- **load threshold *load\_value*** this command is used to decrease to bring the server out of rotation at a lower load value than 254.

```
load threshold 100
```

*!--- When a service hits a load of 100, it  
!--- will be taken out of rotation.*

---

## Related Information

- **CSS 11000 Series Product Support**
  - **CSS 11500 Series Product Support**
  - **Technical Support – Cisco Systems**
- 

All contents are Copyright © 1992–2006 Cisco Systems, Inc. All rights reserved. Important Notices and Privacy Statement.

---

Updated: Jan 20, 2006

Document ID: 12579

---