

Making Cisco CSS 11000 HTTP Keepalives and ColdFusion Servers Work Together

Document ID: 15048

Introduction

[Understanding HTTP Header](#)

[Understanding Differences between HTTP HEAD and HTTP GET Methods](#)

[How ColdFusion Servers Respond to HTTP Keepalives](#)

[HTTP Keepalive Responses the CSS 11000 Understands](#)

[Another Keepalive URI and ColdFusion Issue](#)

[Using Scripted Keepalives as a Workaround](#)

[Related Information](#)

Introduction

A Cisco CSS 11000 series content services switch periodically sends a keepalive message to a service to determine whether the state of the service is alive, dying, or dead. You can use a keepalive as a health check to make sure the service is up so that the CSS 11000 can use the service for load balancing. Otherwise the CSS 11000 removes the service from the load-balancing algorithm. An advanced (Layer 5) type of keepalive is a Hypertext Transfer Protocol (HTTP) that uses an HTTP HEAD (default) or HTTP GET method. A CSS 11000 running software Cisco Web Network Services (WebNS) Software Release 4.10 or later that uses HTTP keepalive expects the response packet in a specific format as described below.

ColdFusion servers (version 4.5.X and later) or JRun servers that run behind a CSS 11000 do not by default respond to CSS 11000 HTTP keepalives in the expected format. You must configure such servers to return specific expected values in response to HTTP keepalives from the CSS 11000. As a workaround, you can use scripted keepalives in the CSS 11000, using WebNS 4.10 and later.

Understanding HTTP Header

The following is a typical HTTP GET request that is made by Web browser:

```
HTTP: ----- Hypertext Transfer          Protocol -----
HTTP:
HTTP: Line 1: GET / HTTP/1.1
HTTP: Line 2: Accept: application/msword,image/gif,image/x-xbitmap, imag
HTTP: e/jpeg, image/pjpeg, application/vnd.ms-excel, application/v
HTTP: nd.ms-powerpoint, */*
HTTP: Line 3: Accept-Language: en-us
HTTP: Line 4: Accept-Encoding: gzip, deflate
HTTP: Line 5: User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Windows NT)
HTTP: Line 6: Host: www.cisco.com
HTTP: Line 7: Connection: Keep-Alive
HTTP: Line 8:
HTTP:
```

A response from the Web server is made up of a protocol version identifier, status code, human-readable

response status line, response headers, and information for the request.

```
HTTP: -----      Hypertext Transfer Protocol  -----
HTTP:
HTTP: Line 1: HTTP/1.1 200 OK
HTTP: Line 2: Date: Mon, 28 Aug 2000 22:52:24 GMT
HTTP: Line 3: Server: Apache/1.3.12 (Unix)
HTTP: Line 4: Transfer-Encoding: chunked
HTTP: Line 5: Content-Type: text/html
HTTP: Line 6: Connection: close
HTTP: Line 7:
HTTP: Line 8: 1000
```

Understanding Differences Between an HTTP HEAD and an HTTP GET Methods

The CSS 11000 HTTP keepalive uses one of two available methods:

- HEAD
- GET

They are specified by using the **method** command while in the keepalive configuration mode (**config-keepalive**).

When an **HTTP HEAD** keepalive is issued in the CSS 11000, a `200 Status OK` is expected in the response packet from the server it is querying. If the `200 OK` status is not returned, the CSS 11000 considers the service down. If the CSS 11000 receives any other status code such as a `404 Object Not Found` or a `302 Object Moved`, the CSS 11000 will view this as an incorrect response and mark this service as down. The head method is the default. The CSS 11000 does not compute a reference hash value for this type of keepalive.

When an **HTTP GET** keepalive is used, the CSS 11000 expects not only to see the `200 Status OK` in the response packet, but it will also run a checksum on the entity body. The first time the CSS 11000 goes out to query, the server will run a checksum on the entity body and store that hash value for future queries. If future responses pass back a `200 Status OK` and the hash value that is calculated is different from the stored value, then the service is considered down. If the `200 Status OK` is not returned, or if the `200 OK` status is returned but the hash value is different from the reference hash value, the CSS 11000 considers the service down.

When you specify an URL for an HTTP keepalive, the CSS 11000 calculates a hash value for the Web page specified in the URL. If the Web page changes, the hash value no longer matches the original hash value, and the CSS 11000 assumes that the service is down. To prevent the CSS 11000 from assuming that a service is down because of a hash value mismatch, specify the keepalive method as the head. Because of the nature of HTTP GET keepalives and the way they are calculated, do not use them on dynamically changing pages because their checksums change constantly.

How ColdFusion Servers Respond to HTTP Keepalives

ColdFusion servers respond differently to HTTP keepalives. A ColdFusion response comes in two packets. ColdFusion usually works like this:

```
ColdFusion Server <-----> CSS 1100
```

- (1)SYN
- (2)SYN ACK
- (3)ACK
- (4)HTTP "HEAD/pages/isrunning.cfm"
- (5)HTTP data (200 OK)
- (6)HTTP data (????????)
- (7)RST,ACK
- (8)RST WIN=0 (????????)

A response packet from a ColdFusion server looks similar to the following:

```
>- - - - - Frame 4- - - - -
48 54 54 50 2f 31 2e 31 20 32 | HTTP/1.1 2
>0040: 30 30 20 4f 4b 0d 0a 53 65 72 76 65 72 3a 20 4d | 00 OK..Server:
>0050: 69 63 72 6f 73 6f 66 74 2d 49 49 53 2f 34 2e 30 | Microsoft-IIS/4.0
>0060: 0d 0a 44 61 74 65 3a 20 46 72 69 2c 20 32 35 20 | Date: Fri, 25
>0070: 41 75 67 20 32 30 30 30 20 31 32 3a 33 32 3a 30 | Aug 2000 12:32:07 GMT.
>- - - - - Frame 5 - - - - -
>0080: 37 20 47 4d 54 0d 0a |
>0040: 70 65 3a 20 74 65 78 74 2f 68 74 6d 6c 0a 0a 3c |
><<-----
>0050: 48 54 4d 4c 3e 0d 0a 3c 48 45 41 44 3e 0d 0a 3c |
>0060: 42 3e 43 4f 4c 44 46 55 53 49 4f 4e 20 5375 63 | COLDFUSION
>0070: 63 65 73 73 66 75 6c 6c 79 20 52 75 6e 6e 69 6e | Successfully Running
>0080: 67 20 3c 2f 42 3e 0d 0a 3c 2f 48 45 41 44 3e 0d |
>0090: 0a 3c 42 4f 44 59 3e 0d 0a 0d 0a 3c 2f 42 4f 44 |
>00a0: 59 3e 0d 0a 3c 2f 48 54 4d 4c 3e |
```

In most cases, you can expect to see the following output:

```
Web Server <-----> CSS 11800
```

- (1)SYN
- (2)SYN ACK
- (3)ACK
- (4)HTTP "HEAD/pages/isrunning.cfm"
- (5)HTML data (200 OK)
- (6)RST,ACK

HTTP Keepalive Responses the CSS 11000 Understands

Packets need to contain a content-length: <value> tag.

After receiving the response header, the CSS 11000 searches for a content-length tag to determine how much data will come in the body. The CSS 11000 monitors the decrementing of this value to determine if it has a complete response. This is imperative in HTTP 1.1 persistent flows. If the response type is HTTP 1.0, the content length is optional. The CSS 11000 will use it if it is present. Otherwise, the CSS 11000 sets the uIndicatedLen to 0 and watches for a connection closure. Lastly, the request method must be considered to determine the existence/nonexistence of an entity body. If the request method was HEAD, there should NOT be an entity body.

The content-type field should be terminated with 0x0d0a<CR><LF> and not <LF><LF>.

The CSS 11000 expects to see the response packet in one packet, not two.

The CSS 11000 can handle a keepalive response that is split across two packets. When the first frame arrives, the CSS 11000 must ensure that it has the entire HTTP response header so it can begin to process the keepalive response.

The HTTP response header should be allowed to span packets and should end in 0x0d0a0d0a <CR><LF><CR><LF>. However, if there are encounters with servers that only responded to the HTTP response header that terminated with <CR><LF>, a CSS 11000 that follows RFC2068 standards will support either. To support fragmentation of the response header and either terminator, if the 0x0d0a <CR><LF> is the segment terminator, meaning they are the last two characters in the packet, then it is assumed that this is the terminator of the HTTP response header.

Cold Fusion servers terminate the first packet (segment) with 0x0d0a <CR><LF>, thus the CSS 11000 assumes this is the end of the HTTP response header and tries to process it. Because the rest of the header is really in the second packet, it fails, which means the service will never come up. Ideally, the response should occur in one packet but if two packets are used the CSS 11000 needs the following:

The first packet cannot be terminated with <CR><LF>. It needs to get all the data combined in one packet or the application needs to ensure that the split between packet 1 and packet 2 does not occur at the <CR><LF>. Data that comes in two packets might encounter issues with how the CSS 11000 will parse the two packets.

Another Keepalive URI and ColdFusion Issue

When a HEAD or a GET is made to a file that does not exist on the ColdFusion server, the server will respond with an HTTP response code of **200 OK** and a **404 Object Not Found** message in the HTML body.

For example, here is an HTTP HEAD request:

```
HTTP: ----- Hypertext Transfer Protocol -----
HTTP:
HTTP: Line 1: HEAD /keepalive.cfm HTTP/1.0
HTTP: Line 2: Connection: Keep-Alive
HTTP: Line 3: User-Agent: Mozilla/4.06 [en] (WinNT;I)
HTTP: Line 4: Accept: image/gif, image/x-bitmap, image/jpeg, image/pjpeg, */*
HTTP: Line 5: Accept-Language: en
HTTP: Line 6: Accept-Charset: iso-8859-1,* ,utf-8
HTTP: Line 7: Host: 216.251.240.75
```

Here is a typical response if the **/keepalive.cfm** file is not present.

```
HTTP:
HTTP: Line 1: HTTP/1.1 200 OK
HTTP: Line 2: Date: Thu, 19 Oct 2000 21:25:51 GMT
HTTP: Line 3: Server: Stronghold/3.0 Apache/1.3.12 C2NetEU/3012 (UNIX) PHP
HTTP: /3.0.16 mod_ssl/2.6.4 OpenSSL/0.9.5a mod_perl/1.22
HTTP: Line 4: Connection: close
HTTP: Line 5: Content-Type: text/html
HTTP: Line 6: Error Occurred While Processing Request
HTTP: Line 7: Error Diagnostic Information
HTTP: Line 8: An error has occurred.
HTTP: Line 9: .0 404 Object Not Found
```

Using Scripted Keepalives as a Workaround

You can use scripted keepalives in the CSS 11000 using WebNS 4.10 and above (any builds). Search for the 404 Object Not Found string in the response. By default, the CSS 11000 will use canned scripted keepalives to check the health of standard services such as Simple Mail Transfer Protocol (SMTP) and Netbios. The CSS 11000 allows customers to customize their own scripts, although they will not be supported by the Cisco TAC. Customers must debug and troubleshoot their own scripts.

Note: Tweak the configuration of other third party application servers to make it work with a CSS 11000 by using HTTP keepalives.

Related Information

- [Cisco CSS 11000 Series Content Services Switches](#)
 - [Content Delivery Products](#)
 - [Keepalive Configuration Mode Commands](#)
 - [Technical Support – Cisco Systems](#)
-

All contents are Copyright © 1992—2004 Cisco Systems Inc. All rights reserved. Important Notices and Privacy Statement.

Updated: Aug 03, 2007

Document ID: 15048
