Workaround for CFHTTP and Compressed HTTP Response from IIS

Posted At : July 28, 2004 7:43 AM | Posted By : Steven Erat
Related Categories: Java, ColdFusion

Some versions of IIS may use an HTTPCompression algorithm that is incompatible with ColdFusion (or Java in general). When using CFHTTP to performthe operations GET or POSTon the target website, if the target site is using Microsoft IIS with HTTP Compression enabled, the CFHTTP tag may report a Connection Failure. If the packets are traced with **Ethereal**, you might even see that all the compressed data is returned to the CFHTTP tag in the response, and the status code might even be 200 OK. If you are getting a Connection Failure result, the problem is that ColdFusion doesn't know how to deflate the compressed content. If this situation occurs, then you can specify additional headers to be sent with the CFHTTP request which tell the target webserver to disable compression for that response.

```
[cfhttpparam type="Header" name="Accept-Encoding" value="deflate;q=0">
[cfhttpparam type="Header" name="TE" value="deflate;q=0">
```

Thanks to Dan Switzer for adding a comment which suggests using an Accept-Encoding value of * instead of deflate;q-0. The * worked for me in situations where deflate did not.

```
[cfhttpparam type="header" name="Accept-Encoding" value="*" />
[cfhttpparam type="Header" name="TE" value="deflate;q=0">
```

The above headers, when used in cfhttpparam, will request that the target webserver send an uncompressed response. The headers use a q parameter to specify the quality factor as described in **RFC 2616**.

I couldn't reproduce this using CFMX 6.1 on Windows XP connecting to IIS 6 on Win2k3 with HTTP Compression turned on for application and static files. Not all versions ofIIS mayhave this problem, and perhaps it was fixed in recent IIS versions. In my case, I confirmed that IIS was sending compressed content by verifying that the compressed output was stored in the IIS Temporary Compressed Files directory and a packet sniff showed the compressed content as gibberish. CFHTTP was able to write out the cfhttp filecontent successfully, and reviewing that output confirmed that the content was properly deflated and readable. Here is that HTTP Request/Response from CFHTTP without using the headers described above.

```
GET /serat/cgi.htm HTTP/1.1
Host: ps-win2k3svr1
Connection: close, TE
TE: trailers, deflate, gzip, compress
```

```
User-Agent: ColdFusion
Accept-Encoding: deflate, gzip, x-gzip, compress, x-compress
```

```
HTTP/1.1 200 OK
Content-Length: 192443
Content-Type: text/html
Content-Encoding: gzip
Content-Location: http://ps-win2k3svr1/serat/cgi.cfm
Last-Modified: Wed, 28 Jul 2004 14:16:51 GMT
Accept-Ranges: bytes
ETag: "80cb9f86ad74c41:4aa"
Vary: Accept-Encoding
Server: Microsoft-IIS/6.0
Date: Wed, 28 Jul 2004 14:34:26 GMT

{lots of encoded data removed here}
```