

The "OK" Button Strikes Again (Cfhttp from CF 6.1 to CF 8.01)

Posted At : February 10, 2009 3:16 PM | Posted By : Mark Kruger

Related Categories: ColdFusion

Recently I migrated a server from ColdFusion 6.1 to ColdFusion 8.01. Typically such migrations are a walk in the park. There are very few things that go wrong. I'll save my check list for another time, but here is one that is new to me. I had some code that utilized the UPS rate checker API. It was fairly straight forward stuff. But for some reason it began throwing a strange error soon after migrating. Here's the scoop.

The Code (Really More Like Guidelines)

The code in question did the following. It looped through some data and created 2 well-formed XML packets. Then it posted them both scrunched up together to the UPS API URL using CFHTTP. Something like this.

```
<cfxml variable="permissions">
<AccessRequest xml:lang="en-US">
<AccessLicenseNumber>...</AccessLicenseNumber>
....
</AccessRequest>
</cfxml>

<cfxml variable="rateReq">
<RatingServiceSelectionRequest xml:lang="en-US">
<Request>
<TransactionReference
...
</RatingServiceSelectionRequest>
</cfxml>

<cfhttp url="#UPSurl#" method="post" name="UPSXMLResponse">
<cfhttpparam type="xml" name="name" value="#permissions##rateReq#">
</cfhttp>
```

On Coldfusion 6.1 this worked splendidly. It returned an XML packet in the file content area which I passed to the xml parser to create an object and then used to set my values for the shopping cart. But on ColdFusion 8 it threw an error. The error was an "invalid variable syntax error" - you know, the kind you get when you try to do `<thisVar = 123xxx/>`. The details said:

The column name "<?xml

```
version=1.0?><RatingServiceSelectionResponse><Response><TransactionReference
....*more*" is invalid.
```

Hmmm.... very interesting (but stupid). The "CFHTTP" return structure was not being created either. My first thought was that for some reason CF 8 didn't like the fact that I was squishing 2 separate xml documents together into my "XML" variable when posting to UPS. I went out and got Ray Camden's code from [Ria Forge](#), but he was doing the same thing - putting 2 packets together on the same line.

The Fix

Finally, I got to thinking about the error. Isn't it amazing how examining the error can be helpful? Some programmer has thoughtfully taken all that time to give you more information and you simply gloss over it like he or she was chopped liver. Of course I'm not alone. Here's a call I get all the time:

- MOM: Hey Son, I got an error when I did x y and z on my computer. Can you Help me?
- MUSE: Sure, what did it say?
- MOM: What did what say?
- MUSE: The error Mom... what did the error say?
- MOM: I dunno, I clicked on OK.
- MUSE: That's ok Mom, try it again and tell me what it says
- MOM: Ok... just a sec... there! It did it again.

- MUSE: Great... what did it say?
- MOM: Oh... I clicked on OK again.

May I just say curse the programmer who ever came up with the "OK" button. I think most users think it's a fix as in "If I click ok this error will be cleared up - right?"

Anyway, not to be outdone by my mom, I was also ignoring the error in this case. The message detail of the error said clearly "the column name blah blah blah... is not valid". Looking closely it was apparent that CF 8 was trying to set the return value from the code (the stuff that is usually found in CFHTTP.Filecontent) into a column name in a query. Why would it want to do that?

Aha! It's because of the NAME attribute of the cfhttp tag. You see when you specify a "name" for a cfhttp call, CF gets the data and returns a query for you. This can be useful in a few cases, but the data at the destination URL has to be formatted to be easily read into a query. In particular it should be comma delimited with the first line being the column names. In this case the "name" attribute causes CF's query parser to kick in. The first thing it does is try to figure out what columns to set up for the return query. Since the first line of the returned content was XML and *not a list of columns*, it tried to take that first line (the string beginning with <RatingServiceSelectinResponse...) and use it as a column name in the query. Since the string is *not* a valid variable name (or column name in a query) the code threw an error and did not follow through and create the CFHTTP structure.

Why Is it Interesting?

Now, what is useful to understand here is that this is one of those really nuanced behaviors that you *might* run into when moving from CF 6.1 to CF 8. After all, the code in question did *not throw an error* in CF 6.1. Instead, it simply glossed over the query creation and gave back the CFHTTP.filecontent. Since I was not using the return as a query anyway, I did not actually need the "name" attribute. I assume it was left over from a previous way of calling the UPS service (this is legacy code, about 6 or 7 years old).

So if you see a funky "column name" type error being thrown from a CFHTTP call and you have just moved from CF 6.1 to CF 8 - this could be your issue. Happy troubleshooting!