

Examining isDefined()

Posted At : August 6, 2009 1:38 PM | Posted By : Mark Kruger

Related Categories: ColdFusion

While perusing one of my email lists I stumbled onto a behavior of the "isDefined()" function that bears repeating. This function is commonly used in *most* ColdFusion applications. In fact, I would put it in the top 10 of functions used (perhaps the top 5), so any bug or interesting behavior related to isDefined() should warrant some notice. The short description of the problem is that isDefined() *may* throw an exception during a lengthy request. The conditions have to be just right.

If you are wondering if this behavior is related to an error you are experiencing, one clue is in the exception information. If you are seeing something like "*error Error while reading header [VARIABLE]*" in conjunction with a socket write error (connection reset by peer: socket write error) then you should probably take a closer look at this post.

But before we discuss the behavior, it's important to understand how "isDefined()" works. As you know everything in ColdFusion belongs to some scope or is a member of some object. So ColdFusion has to work it's way down an order of precedence when trying to figure out if something is defined or not. Something like this.

Dispatcher and Worker Thread Have a Party

- **Dispatcher:** Yo Engine I got one a dem isDefined things here.
- **Worker Thread:** Uh... what's with the New York cabbie accent. You know you are from San Francisco right?
- **Dispatcher:** I know I know... it's the name they give me "dispatcher". I can't help myself. I've even started smoking cigars and scratching myself in...
- **Worker Thread:** [interrupting] Whoa there big fella... that is waaay more information than I need. How about we take a look at the task at hand before you waste another millisecond of my life I'll never get back. I only live till about 1000 or so you know.
- **Dispatcher:** [a little miffed] Fine... I have an isDefined call for you. Can you check to see if the variable "giJoe" exists.
- **Worker Thread:** giJoe eh.... let's see, giJoe... giJoe [humming softly and muttering]... Is it a local function in a CFC? Nope. Am I a spawned thread? Nope. giJoe....giJoe.... Is it in an arguments scope? Nope. Is it part of the Variables scope? Nope. Is it in a thread scope? Nope... when was the last time I saw that anyway? ... hmmm... is it in the CGI scope? Nope. Is it in the cfile scope? Nope. Is it a part of the URL scope? Nope. Is it part of the form scope? Nope. How about a cookie? Sounds delicious but nope (little humor there dispatch...)
- **Dispatcher:** Hilarious. You should be on the circuit - oh wait, you *are* on a circuit [snicker].
- **Worker Thread:** ...um... Client scope? Nope. Well Dispatch, it looks like I have to say false - this variable is not defined anywhere.
- **Dispatcher:** Great, I'll pass that along. See you in a few million iterations.

What our little play illustrates is that ColdFusion has to look through a complete list of scopes when using isDefined(). In fact, isDefined() takes no notice of your dotted

notation unless the first part of the string *matches the scope it is examining at the time*. Time for an Example:

```
<Cfset variables.form.giJoe = "It's not a doll"/>

<cfoutput>#isDefined('form.giJoe')#</cfoutput>

<cfif isDefined('form.giJoe')>
    <cfoutput>#form.giJoe#</cfoutput>
</cfif>

<cfset form.gijoe = "It's an action figure"/>

<cfif isDefined('form.giJoe')>
    <cfoutput>#form.giJoe#</cfoutput>
</cfif>
```

Let's step through this code. First, we set a variable in the variables scope called "form.gijoe". When we first check "*isDefined('form.giJoe')*" the function returns *TRUE*. Why? Because the variables scope is checked before the form scope. In fact, *isDefined()* goes through a ginormous lists of scopes checking each one for the existence of *form.giJoe*. They go in this order (this is from the CF 8 docs).

1. Function local (UDFs and CFCs only)
2. Thread local (inside threads only)
3. Arguments (like when you are inside of a function)
4. Variables
5. Thread (as in *thread.this* or *thread.that*)
6. CGI
7. Cffile
8. URL
9. Form
10. Cookie
11. Client

Secondly, the first *cfoutput* will output the values of *variables.form.gijoe* - even though from a readability standpoint we likely *think* the code is outputting something from the form scope (which is one reason to never use *variables.form* or *variables.url*). Finally, the second *CFIF* (the one that occurs after we have set a "gijoe" into the form scope)...

```
<cfif isDefined('form.giJoe')>
    <cfoutput>#form.giJoe#</cfoutput>
</cfif>
```

...returns true, but *not because there is a giJoe in the form scope*. Indeed, it is returning true for the same reason it returned true a few lines above. It has found *form.giJoe* in the *variables* scope. It hasn't even gotten to the form scope yet. Are you confused? Me too. Let's see if, given all we know so far, we can draw some conclusions.

isDefined() is More Expensive Than StructKeyExists()

Although its been said many times many ways (with my apologies to Johnny Mathis) it bears repeating that *isDefined()* will not perform as efficiently as *structkeyExists()*. *StructKeyExists()* has an easier time of it because the code must identify a scope or structure for the function to examine. For *isDefined()* to work you throw it some string and say "figure it out and get back to me" - which naturally takes more steps and is

more time consuming.

Dotted Notation Nuances

Secondly, when you use dotted notation (i.e. `isDefined('form.giJoe')`), the left hand side only matters if the parser gets around to that scope. For each scope it examines `isDefined` takes a peak at the left hand side and determines if it's the same name as the scope in question. So simply doing something like this:

```
<cfset form.RonWeasley = "Is Our King"/>
<cfif isDefined('form.RonWeasley')>
    ...
</cfif>
```

... will work successfully. When the parser gets to the form scope it will match up with the left hand side and say "aha!" - and start looking for the right hand side of the dotted notation within that scope. In fact, when it stumbles upon the scope you had in mind it is functioning exactly like `structKeyExists`.

And Now For Something Completely Different

Now back to our error. Remember I said it had to be a lengthy request. Indeed for this particular error to crop up, the requests needs to be running long enough for the connection between JRUN and the web server to be dropped. So, for example (and this is pure conjecture), say a database query hangs for a long time and the web server gives up drops the connection to JRUN. JRUN is still drumming its fingers (or "digits" I suppose) and when it finally gets the query return from the DB server it moves to the next task - an `isDefined()` call. Remember our order of precedence?

When the `isDefined()` routine gets to scope number 6 in its order it is going to try and access the CGI scope. The CGI scope is contained in a reference or header - something provide by the web server... the *web server that has now moved on to bigger and better things*. When it tries to sort through the CGI information it can't "read" from the information that it expected to be there because the web server no longer has an open connection with this particular request. So it throws the sort of exception that you might expect - a "socket error" (Connection reset by peer: socket write error) and an "Error while reading header" error. Jrun is trying to send a message to the web server so it can examine the CGI scope, and it fails because the socket it closed or the memory dereferenced or whatever.

Takeaways

I'm told this condition has been around since CFMX. It seems like something that *could* be addressed by a fix to the connector, but I will leave it to Adobe to figure out if that is possible or not. Such an error will affect servers doing a lot of heavy lifting where there are potentially long running requests. Of course, many folks (myself included) will say that `isDefined()` should be used quite sparingly and that `structKeyExists()` is the right way to go 99 percent of the time. Still, a host of legacy code and procedural programming that is still in the wild means we will be dealing with `isDefined()` issues for the foreseeable future. So Adobe if you are listening, it makes sense to fix this issue.

When to Watch For It

You might see a flurry of these sort of errors after something has managed to hang or

queue a group of threads on your ColdFusion server. For example, if your DB is having performance issues it could result in queuing threads - threads waiting for the DB to return. When the threads start clearing out, some percentage of the JRUN connections to the web server will likely have been closed - resulting in errors *if* isDefined() is subsequently called. If you have had such performance issues check your log files. If you see a plethora of socket write errors you should consider some code refactoring to get rid of the offending isDefined() errors.

My thanks to CF Gurus [Cameron Childress](#) and [Sean Corfield](#) for helping to explain this behavior.