

Setting Timeout Successfully on a Web Service Call

Posted At : March 21, 2013 2:17 PM | Posted By : Mark Kruger

Related Categories: Coldfusion Tips and Techniques

One of the annoying things about ColdFusion (yes even the Muse gets annoyed) is the sort of haphazard way it deals with timeouts. If the process you are timing out involves a call to an external service it's really a crap shoot whether or not it will work. Once CF hands off to the external service and starts its vigil waiting for the callback, the timeout value is largely ignored. Don't believe me? Create a long query to a DB Server and then pull the network cable while the query is running. The thread will usually continue to hang even if you have added a timeout value.

Recently Super Guru Jared Riley ([Computer Services Inc. \(CSI\)](#)) was lamenting this very problem with regard to a web service he was using. Because the web service would sometimes hang at the other end due to reliability issues his server was accumulating dormant ColdFusion threads which eventually would fill up the simultaneous request pool and begin to queue all other requests - effectively locking up the server.

It turns out a savvy developer named Jeff Nelson (also of CSI) came up with a solution for this particular issue. Before I share I must warn my readers that this is an *undocumented* solution that sets an underlying AXIS property. That means that subsequent changes to some future version of the underlying Axis libraries could cause this to error out at some point. The Muse has been known to use undocumented features successfully from time to time - but it pays to be vigilant when upgrading or patching. Also keep in mind this is *only for web services*. It will do nothing for Queries or cfmail etc. With that in mind here is the "fix".

Setting the timeout axis property

```
<cfset webservice =  
createObject('webservice', 'http://exampledomain.com/webservice.cfc?wsdl') />  
<cfset webservice._setProperty("axis.connection.timeout",  
                                javaCast("int",10000)) />
```

Now some of you might immediately say "hmmmm.... that's the connection timeout, but it doesn't really cover long running requests that occur AFTER the connection is made does it?" Jared has actually done a good bit of testing and claims that this property *will* timeout a request for either a connection reason or a time of process reason.

So if you are trying to solve this particular problem this might be an appropriate course of action. Now if we could just find similar settings for various DB Drivers my life would be complete.

Follow Up

For those of you who want to remind me that there already *is* a timeout property to cfinvoke that can be used here I would respond that that setting works correctly for *creating the stub classes*. In other words if ColdFusion can't compile the WSDL with the time allotted it will timeout. But it doesn't work for actual calls to the methods instantiated.

