

# Choices for Client Vars: Registry Bad, Datasource Good

Posted At : October 27, 2009 12:58 PM | Posted By : Mark Kruger

Related Categories: ColdFusion

## Client Variables and the Registry

Ask any experienced ColdFusion troubleshooter and he will tell you the same thing, *"Don't store client variables in the registry."* In fact, when examining a sick server one this is one of the first items I look at. If the customer says "It seems like the server stops about every hour" it's a safe bet that the default storage is set to *Registry* and the default purge interval has been left alone at 1 hour and 7 minutes (which is kind of an odd interval - probably some Adobe employee's anniversary in binary).

In many cases this is a "hidden" problem waiting to burst onto the scenes and bite some poor site owner in his nether regions. The owner launches his or her site and begins to gather traffic with the default settings for client variables. By default ColdFusion stores 90 days worth of client variables in the Registry - so the site can actually perform well for a few months. But then, out of no where, the server starts to drag and even stop every hour or so. Under the hood the purge operation is starting to find client vars that are 90 days old or more and it is taking quite a long time to delete them. The OS sees the registry keys being deleted and (sometimes) attempts to shrink the registry size. This affects a sort of "locking" on the registry where new keys are not being written - meaning requests are queuing and the server is slowing to a crawl. Now you might think that fixing this is as easy as switching from the registry to a datasource or cookie storage as the default, but there are some nuances to this fix that bear mentioning.

## What about Linux

Before moving on let's talk about Linux. Linux, having nothing really like a "registry" uses a file located in `/coldfusion8/registry/cf.registry`. ColdFusion writes client vars as registry-like entries into this file. In fact you should know that this file is an attempt to *mimic the windows registry*. The file contents look like this:

```
1: HKLM:1
2:
3: HKLM\SOFTWARE:1
4:
5: HKLM\SOFTWARE\Macromedia:1
6:
7: HKLM\SOFTWARE\Macromedia\ColdFusion:1
8:
9: HKLM\SOFTWARE\Macromedia\ColdFusion\CurrentVersion:1
10:
11: HKLM\SOFTWARE\Macromedia\ColdFusion\CurrentVersion\Clients:1
12:
13: HKLM\SOFTWARE\Macromedia\ColdFusion\CurrentVersion\Clients\338751..
14: cfid: 338751; REG_SZ
15: cftoken: CD188FEB-F93F-00EA-FC626B67A1CD00A4; REG_SZ
16: hitcount: 3; REG_SZ
17: timecreated: {ts '2009-10-27 07:35:48'}; REG_SZ
18: lastvisit: {ts '2009-10-27 07:35:50'}; REG_SZ
19: testSQL::this: yes; REG_SZ
20: testSQL::that: no; REG_SZ
21: urltoken: CFID=338751&CFTOKEN=CD188FEB...; REG_SZ
```

If you have ever gone poking around in the windows registry it should look familiar to you. Note, I've replaced the string `HKEY_LOCAL_MACHINE` with `HKLM` to save space above. When client vars are stored in the "registry" for a Linux box, the values are written in keys like the one at line 13 above. In it's default configuration the Global variables are "on". That means in an application with client vars enabled the "hitcount" and "lastvisit" values are updated with *every request*. Now ColdFusion actually holds

on to vars in memory and flushes them to the file periodically. Additionally, Linux is *very* good at handling file reads and writes (it runs circles around 32 bit windows). But even so it should be obvious that on a very busy server and with 90 days worth of these variables this file could become the source of I/O bottlenecks as CF and Linux try to negotiate reads and writes to it.

## I Don't Think I Use Client Vars

"Ah..." you say, "but I never set any client vars." Well if you set the "clientmanagement" key in your Cfapplication tag (or application.cfc properties) - then you are using client vars whether you know it or not. Even if you don't "set" any specific client variables ColdFusion will still create a token for each user and set the "Hitcount" and "lastvist" variables. In fact, if you make no changes to the default configuration, these 2 values are set still written to with *each new page request*.

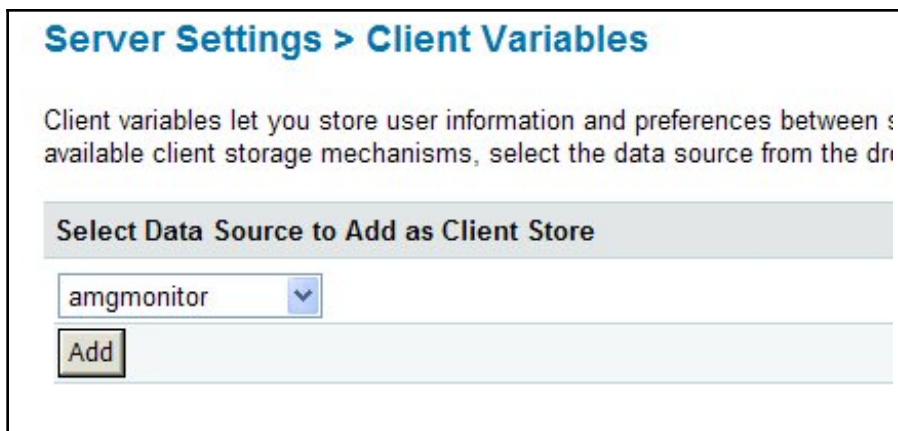
## The Fix

Ok, so why belabor the point. Registry *bad* datasource *good* - right? Hmmmm.. one thing to keep in mind is that not all datasources are created equal. For example, if you move your client vars to an MS Access DB it is probably better than the keeping them in the registry. But that is sort of like saying Adam Sandler is a better actor than Chevy Chase or Windows ME was better than Windows 98. Do yourself a favor and use a real honest-to-goodness RDBMS for your client vars. MySQL or MSSQL are excellent choices (and there are several others).

## The Steps

Before you go into the client vars management app in the ColdFusion Administrator you must first create the datasource to use as your client variable storage DB. This is done through the "datasources" application in the Administrator. We won't go into that here except to note that the datasource must have read, select, insert, update, delete and *create* permissions enabled in order to set it up initially. After you set it up and the tables are created you can drop the "create" permission.

Next, click on the "client variables" link in the left hand admin menu of the CF Administrator. You will see the default list of places to store client vars. Initially you will see just "cookies" and "registry" with "registry" selected. At the top you will see a list of all the datasources you have setup in ColdFusion including the one you just configured.



**Server Settings > Client Variables**

Client variables let you store user information and preferences between sessions. To manage client variables, select the data source from the dropdown menu.

**Select Data Source to Add as Client Store**

amgmonitor ▼

Add

Select the datasource you prepped and click "Add". The next screen shows your configuration options.

**Description**

☒ **Create Client database tables**

Enable this option only if this is the first time you are configuring the cur

☒ **Purge data for clients that remain unvisited for**  **days**

Enable this option if you want ColdFusion to periodically purge client dat

If this data source is being used by more than one ColdFusion server, as

☒ **Disable global client variable updates**

This option controls how ColdFusion updates global client variables, suc  
ColdFusion updates global client variables for each page request.

This screen has one choice that you should consider carefully - the checkbox for *disable global client variables*. These are 2 values that are a part of each set of client vars. By default every time the same client (based on the cfid/cftoken or the JsessionID) hits your page the lastUpdate key is reset and the hitcount key is ticked up by one. Disabling this constant barrage of updates can save your DB from a lot of hits. In fact, as a rule of thumb, unless you are actually using hitcount or lastupdate somewhere in your code you should *always disable* global updates. Why bombard your DB with values you are not going to use? I also usually set the purge value down to 10 days - although this depends greatly on the intended use.

Clicking "submit changes" will create the tables on your DB and take you back to the list. The final step is to select the radio for your new datasource and click Apply.

Actions		Storage Name
<input type="radio"/>		Cookie
<input checked="" type="radio"/>	 	digbroclientvars
<input type="radio"/>	 	edv
<input type="radio"/>	 	necfug
<input type="radio"/>		Registry
<input type="radio"/>	 	scv
<input type="radio"/>		None

Apply

**Why Am I Still Locking Up?**

Have you solved your problem? Perhaps not yet. Remember that the list of client storage options represents all the ways that any application on the server *can use to store client variables*. When ColdFusion goes to purge variables it is going to check out *each* of the choices on the list (except cookie) and attempt to delete keys that are older than *N* number of days. Simply moving to a datasource solves the problem of where these vars are stored *going forward* but if you have had your server in service for a while your registry may still have a gazillion registry entries in the "clients" key. In spite of your fancy pants new database for storage CF still cycles through *all* of the choices in the list - including the registry. Until all those keys are out of there the server might continue to lag.

At least one solution is to simply delete the clients key from the registry (or delete the cf.registry file and replace it with a blank file). The key is located at HKLM\SOFTWARE\Macromedia\ColdFusion\CurrentVersion\Clients. Under no circumstances should you try to left click on this key to load the sub keys. It could have hundreds of thousands or even millions of values in it. I usually stop ColdFusion and then using regedit I right-click on the word "Clients" and delete it the whole key along with it's sub keys. Even this can take a long time. To be safe I recreate a blank "Clients" key in its place and restart ColdFusion. Now, theoretically, ColdFusion has nothing to purge in the registry.

## Hold Your Horses We are Not Done

Recently I fought this client-vars-in-the-registry battle on a Windows 2008 server. The server continued to have problems after I deleted the clients key and moved the client vars to a datasource. I suspected that client vars were not really my problem, but I wanted to eliminate it as a possibility. Windows 2008 server is extremely role and permission conscious and I still suspected that CF's hourly access of the registry was causing problems of some sort. My suspicions were further confirmed when, after setting the purge interval to 18 hours, the server bogged down on cue when the purge operation started (even though there were no values in the registry and no values in the datasource I had created). So I went searching for a way to exclude the registry altogether from the purge operation.

This led me to one of those XML files curiously named by Macromedia developers who can't seem to divest themselves of pop-culture references even at work - the neo-clientstore.xml file (found in Coldfusion8/lib). This file is a WDDX packet containing the list of options for client storage. On careful examination I found this node:

```
<var name='Registry'>
  <struct type='coldfusion.server.ConfigMap'>
    <var name='timeout'>
      <number>90.0</number>
    </var>
    <var name='description'>
      <string>System registry.</string>
    </var>
    <var name='type'>
      <string>REGISTRY</string>
    </var>
    <var name='name'>
      <string>Registry</string>
    </var>
```

```

    <var name='purge'>
      <boolean value='true' />
    </var>
    <var name='disable_globals'>
      <boolean value='false' />
    </var>
  </struct>
</var>

```

As an experiment I commented out this node. I also found a node at the bottom that looked like this:

```

<struct type='coldfusion.server.ConfigMap'>
  <var name='PURGE_INTERVAL'>
    <string>1:7</string>
  </var>
  <var name='default'>
    <string>Registry</string>
  </var>
  <var name='uuidToken'>
    <boolean value='false' />
  </var>
</struct>

```

Naturally I had to change the "default" to something other than registry before removing the registry node. I set it to "Cookie". In fact, take care of changing the default to a datasource before you try to editing this file. That way you won't have to fiddle with the ConfigMap node. After removing the Registry node and restarting I logged into the CF Admin. Registry was no longer listed as one of the options. ColdFusion started and ran successfully.

NOTE: If you do this and any (foolish) developer has actually specified "registry" in the cfapplication tag (or the properties of the Application.cfc component) - as in:

```

<cfapplication name="blah"
  clientmanagement="yes"
  clientstorage="Registry">

```

...then removing the registry node from the neo-clientstore.xml file will *break their code*. Also note, I am still testing to see whether this resolves my problem. I suspect that ColdFusion uses this file to determine what datasources (in this case "Registry" is a datasource for client vars) to purge. I am further assuming that without the registry node in this file ColdFusion will not attempt to access the registry at all - at least that is my hope. I am running a test server configured this and I will report with a follow up post if this has no effect, causes other issues or errors, or is otherwise a poor solution.

Meanwhile, if it pans out I might implement it across the board for all servers - especially windows servers.