

Ask-A-Muse: How Do I Use Client Variables?

Posted At : January 23, 2007 1:22 PM | Posted By : Mark Kruger

Related Categories: Coldfusion Optimization, Coldfusion Tips and Techniques

Muse Reader Asks:

We have a client variables database that is currently storing over 1.5 million records. In the Coldfusion Administrator, client variables are set to purge every 90 days. I was thinking of purging every 7 to 30 days to reduce the number of database entries and hopefully improve performance. What are my options?

It's a good question. Performance depends on many thing including hardware, traffic, tuning, and capacity. Still, 1.5 million rows is a large table and as you might imagine it will perform less ably than a table with just a few hundred thousand rows. Let's talk about these variables and how they are used.

How Long Should I Store Client Variables?

First, what is being stored in this table? Hopefully you know your application well enough to already be aware of what is being stored. Ask yourself how important it is to keep that information. How important is it to personalize repeat traffic? Is it a shopping cart? Contact information? Page customizations? The type of data you are storing as "client" variables will make a difference in this decision. Please not, it is rare to need to store client variables for 30 days. If you need that sort of persistence, then store a cookie and re-instantiate the client variables when the user returns.

Multiple Data Stores

Perhaps one application needs 30 days of storage and the others can get by on just a couple of days. Try this approach. Add another data source as a client variable store using the admin tool. Careful, it will become the default when you add it and you will need to "reset" the default back to your original data store. Your goal here is just to get it "on the list". You can set purge and retention properties on a per database basis. Once you have it added, go to the application that needs the 30 day retention and use the "clientstorage" attribute to specify your other database:

```
<cfapplication
    name="myApp"
    clientmanagement="Yes"
    clientstorage="myNewClientDsn">
```

Now, client variables for this site will be stored in this new DSN while all the other client variables will be stored in the default DB.

Disable Global Variables

If you are not using the "hitcount" or the "lastvisit" variables that are updated automatically by Coldfusion then you should disable this feature (there's a checkbox for "disable global client variables". This will mean that Coldfusion will not need to update the database with every single request made by the client.

Take Care with Client Variables

Maybe you have some code in your application.cfm page like this:

```
<cfif NOT isDefined('client.whoami')>
    <cfeset client.whoami = 'guest' />
</cfif>
```

Simple right? If there is not an "explicit" name or variable indicating that this client is someone we "know" we set a client variable to "guest" - and we treat them as a guest until we know who they are. Behind the scenes Coldfusion creates a new token (cfid/cftoken) and stores it in the CDATA table. Go check it out. You will see the string "whoami=guest" in the data column. So this is ok, right?

You may not realize it but you are setting up to receive a lot of "guest" data in your CDATA table. You see, in order to match up the "client" variable with the data in the CDATA table Coldfusion needs either a cookie (the most common) or a URL parameter with the CFID and CFTOKEN attached. When a user "first" arrives at the site it creates a unique cfid and cftoken and stores it in the table, and also sets it as a cookie on the user machine - or, if you are using the URL method your code must be configured to pass it with every subsequent request.

Now consider all of the "non-human" users that come to your site - google-bots and yahoo and crawlers (oh my!). If you have a busy site with lots of content it will be visited quite frequently by these busy little bees. When a crawler or a bot makes a request it takes the content it finds and extracts URLs from it for indexing. Then it goes through these URLs one by one and indexes the content from them - looking for new content. The catch is that bots don't pass back cookies (or often URL variables). So each request will appear as if it were a new "guest" user coming to visit. The result? The code above will create a new row in the database for every hit by a bot. This has the potential to greatly tax your database so keep it in mind when developing a strategy.

Final Notes

Many developers use client variables as a proxy for session variables. I've seen schemes that push complex objects into WDDX and then store them in the database as client variables. Don't be afraid to use sessions when they are appropriate. Usually the problem with session variables is related to a badly tuned JVM and inappropriate settings - not any specific weakness related to sessions. If all you are doing is tracking information tied to a specific user after login or during a visit to your site then sessions may be a much better choice. At least you will not be dependent on the database.

Also keep in mind that the size of the variable you are setting now has an impact on the process. If you are going to serialize and object into an 8k string of data and store it as a client variable, that data is going to be ported back and forth between the db server and the web server constantly. Finally, and I know this breaks your heart for me to say it, but Access is not a "database server". Do not store your client variables in Access. If the muse catches you doing it he's gonna' slap your hand and take away your birthday! Use a full fledged DB platform like MS SQL, MySQL, or Oracle.