ChatScript MongoDB

Revision 8/27/2016 cs6.8a

ChatScript ships with code and WINDOWS libraries for accessing Mongo but you need a database somewhere. On windows mongo access is the default. On Linux you have to make mongoserver. On Mac and IOS #define DISCARDMONGO is on by default.

If you have access to a Mongo server remotely, that's fine. If you need one installed on your machine, see the end for how to install one.

Using Mongo as file server

Aside from using Mongo to store data for a chatbot to look up, one can also use Mongo as a replacement for the local file system of the USERS directory. This allows you to scale CS horizontally by having multiple CS servers all connecting to a common Mongo DB machine so a user can be routed to any server. Log files will remain local but the `USERS/topicxxxxx` will be rerouted, as will any `^export` or `^import` request. If you name your file `USERS/ltm-xxxxx` then that will be routed to its own collection, otherwise it will go into the same collection as the topic file.

To set collections, on the command line use:

```
mongo="mongodb://localhost:27017 ChatScript topic:MyCollection"
```

or

```
mongo="mongodb://localhost:27017 ChatScript topic:MyCollection ltm:MyLtm"
```

or whatever. Obviously put the correct data for your mongo machine. CS will store user topic files in the Mongo machine and well as `^export` and `^import` data. For user and server logs it will continue to store those on the local machine. You may have the same or different collection active, one or two via filesystem replacement and once via normal script access.

Access A Mongo Database

There are only a couple of functions you need.

`^mongoinit( server domain collection )`

Once the named collection is open, it becomes the current database the server uses until you close it no matter how many volleys later that is. Currently you

can only have one collection open at a time.

By the way, if a call to `^mongoinit` fails, the system will both write why to the user log and set it onto the value of `$$mongo_error`. `^mongonit` will fail if the database is already open.

`^mongoclose()`

loses the currently open collection.

`^mongoinsertdocument(key string)`

does an up-sert of a json string (which need not have double quotes around it. Key is also a string which does not require double quotes.

`^mongodeletedocument(key)`

removes the corresponding data.

`^mongofinddocument(pattern)`

finds documents corresponding to the mongo pattern (see a mongo manual).