MDB2 disconnects and forgets charset setting when reconnecting

Asked 16 years, 3 months ago Modified 16 years, 3 months ago Viewed 2k times

Part of PHP Collective

0

We recently debugged a strange bug. A solution was found, but the solution is not entirely satisfactory.

We use IntSmarty to localize our website, and store the localized strings in a database using our own wrapper. In its destructor, IntSmarty saves any new strings that it might have, resulting in a database call.

We use a Singleton instance of MDB2 to do queries against MySQL, and after connecting we used the SetCharset()-function to change the character set to UTF-8. We found that strings that were saved by IntSmarty were interpreted as ISO-8859-1 when the final inserts were made. We looked closely at the query log, and found that the MySQL connection got disconnected before IntSmarty's destructor got called. It then got reestablished, but no "SET NAMES utf8" query was issued on the new connection. This resulted that the saved strings got interpreted as ISO-8859-1 by MySQL.

There seems to be no options that set the default character set on MDB2. Our solution to this problem was changing the MySQL server configuration, by adding

```
init-connect='SET NAMES utf8'
```

to my.cnf. This only solves the problem that our character set is always the same.

So, is there any way that I can prevent the connection from being torn down before all the queries have been run? Can I force the MDB2 instance to be destructed after everything else?

Turning on persistent connections works, but is not a desired answer.

PHP php mysql pear mdb2

Share Improve this question Follow

asked Sep 17, 2008 at 8:17

Vegard Larsen

13k ● 14 ● 60 ● 102

1 Answer

Sorted by: Highest score (default)

1

✓

From the PHP5 documentation:

> The destructor method will be called as soon as all references to a particular object are removed or when the object is explicitly destroyed **or in any order in shutdown sequence.**
>
> [PHP documentation](#)

(emphasis mine)

What is probably happening is that your script does not explicitly destroy the object, and so when PHP gets to the end of the script it starts cleaning up things in whatever order it feels like--which in your case, is closing the database link first.

If you explicitly destroy the IntSmarty object prior to the actual end of the script, that should solve your problem.

Share

Improve this answer

Follow

edited Jun 20, 2020 at 9:12

Community Bot

1 ● 1

answered Sep 18, 2008 at 5:47

Nathan Strong

2,400 ● 13 ● 17