LiveCode Server

by David Williams on March 6, 2014 5 comments

Share on Facebook

Share on Twitter

Here at On-Rev, we make extensive use of LiveCode Server internally for scripting – you'll see a lot more filenames ending in .lc than you might in .sh, purely for the reason that LiveCode Server is great for making scripts that are a lot more readable than, say, a bash script, and we can drop in shell commands with ease where necessary.

```
#![redacted]/livecode-server

put word 2 of shell("ps aux | grep 'backupscript.lc' | grep -v grep") into tPID
put logText("LIMITING CPU TO 80") & return
put shell("screen -d -m cpulimit -p" && tPID && "-l 50") & return

if $0 is empty then
	set the folder to "/home"
	put the folders into tAccounts
else
	put $0 into tAccounts
end if

replace "." with empty in tAccounts
replace "/" with empty in tAccounts

repeat for each line tAccount in tAccounts
	if tAccount is empty then next repeat
	put logText("RSYNCING HOMEDIR DATA FOR" && tAccount) & return
	put shell("rsync -avz /home/" & tAccount && "[redacted]") & return
	put logText("FINDING SQL DBS") & return
	put shell("/bin/ls -1 /var/lib/mysql | grep '" & tAccount & "_'") into tDBList
	put logText("FOUND DBS:") & return & tDBList & return
	put logText("DUMPING") & return
	repeat for each line tDB in tDBList
		put shell("mysqldump" && tDB && "| ssh [redacted] 'cat > [redacted]/MYSQL_DBS/" & tDB & "'") & return
	end repeat
	put logText(tAccount && "DONE, MOVING TO NEXT") & return
end repeat
put logText("BACKUP DONE")

function logText pText
	return "----------" & pText & "----------"
end logText
```

One recent feature that was introduced in LiveCode 6.6 was the ability to use hashbangs (#! /path/to/livecode) in LiveCode server scripts instead of script open and close tags (<?lc ?>). This means that we can have scripts which look like this:

#!/path/to/livecode-server

put "hello world at" && the millisecs

Instead of this:

<?lc

put "hello world at" && the millisecs

?>

The difference is that the latter has to be passed as a file to the

LC Server exectuable, which then parses the code for output. This makes sense where LC Server is integrated into a web server software for serving webpages, but if we were doing this in a command line, it would look like this:

```
root@mybox [~]# /path/to/livecode-server ./script.lc
```

Whereas by using hashbangs, we can now execute the script directly, as the script contains the information for how it should be executed:

```
root@mybox [~]# ./script.lc
```

This seems like a minor change, but it brings the LC Server engine closer to how scripting for system utilities should function in a Unix environment, and allows me to tidy up some of the internal systems we use a little bit.

Additionally, work on the On-Rev client has been ongoing and we hope to release an update as soon as some of the last technical hurdles are overcome – we hope to send out a notification about this to all our On-Rev customers in the near future.