

# LiveCode for Developers

Creating release quality applications with LiveCode

## Using the livecode standalone engine to run script only stacks on a server

Years ago I set up a license key generator on our server to generate license keys for our products. I used a revolution standalone engine and a plain text LiveCode script that was processed using the engine. Recently, a server upgrade removed a file that was necessary for the old revolution engine to function so I was forced to update my setup. Below is a description of the solution I came up with after a few hours of trial and error.

### LiveCode Community Standalone Engine

To process my plain text LiveCode script files I installed the 64-bit version of the community version of the Linux standalone engine on my server. I grabbed the file named Standalone from the **Tools/Runtime/Linux/x86-64** folder of a LiveCode Community installation. I renamed the file to `<code>` on my server and made it executable (e.g. `chmod 755`).

Note that you need to use the community version as it doesn't need to be licensed. If you grab the Standalone engine from an Indy or Business installation the engine won't run because it is unlicensed. Licensing occurs when you build a standalone in LiveCode.

### The Script

For the script I created a **Script Only Stack** which looks like this:

```
script "clarifykeygenerator"

on startup
  put generateKey(the commandArguments)
  quit
end startup
```

```
function generateKey pParams
    ...
    return tKey
end generateKey
```

The script generates the key, outputs it to STDOUT, and then quits. That's it.

You can now test this setup on the command line. Assuming the **lc-standalone-v8** engine and a **keygenerator.livecode** script are in the same directory the following call would work:

```
./lc-standalone-v8 -ui keygenerator.livecode theLicenseCount theLicenseName
```

Note that when processing parameters in your script, **-ui** and **keygenerator.livecode** would be the first two parameters. Actual parameters that your script is interested in processing would start with parameter number 3 in the `commandArguments`.

## Calling The License Generator From PHP

The final thing I had to do was call the license generator from PHP. You can do that using the **exec()** function. Below is some sample code. Note that I am base64 encoding the name. If the name for the license has accentuated characters in it then PHP will throw an error if you try to include it in the **exec()** function. Just make sure you call **base64decode** in your LiveCode script that generates the license key.

```
$license_generator = '/usr/home/name/public_html/cgi-bin/';

$execStr = $license_generator . 'lc-standalone-v8 -ui ' . $license_generator .
    'keygenerator.livecode ' .
    escapeshellarg($license_count) . ' ' .
    escapeshellarg(base64_encode($name));

$key = exec($execStr, $output, $return_var);
```



This entry was posted in Uncategorized on October 21, 2016 [<https://www.bluemangolearning.com/live->

code/2016/10/using-the-livecode-standalone-engine-to-run-script-only-stacks-on-a-server/]

---

One thought on “Using the livecode standalone engine to run script only stacks on a server”



Tom Bodine

October 24, 2016 at 11:29 am

An eye-opener! I'd never considered using a standalone engine in this manner. Thanks for the insight.

---