

Tired of the permission errors with your cache/logs in a Vagrant environment? Me too!

After being mildly annoyed with having to manually delete `/var/cache` and `/var/logs` repeatedly during “local” development, I decided to brut force matters into my own hands and solve this little annoyance once and for all. Ironically, [Symfony 3.3 is addressing some of this](#), so it may be obsolete by then, but in the meantime...

What are we doing here?

1. Define cache and logs directories for a specific environment.
 1. Wouldn't it be awesome if we could customize these as parameters in the `FrameworkBundle`? Yes, yes it would!
2. Write a simple `CacheCommand` that overrides the default `cache:clear`
3. PARTY!!!!

Customize your `getCacheDir` and `getLogDir` methods by your environment

In my case, I typically use Vagrant/Virtualbox for local development, and Docker containers/bitbucket-pipelines for “test” while Production is a variety/flavor of AWS EC2 instances which have some additional flare.

```
//app/AppKernel.php
```

```
public function getCacheDir()
{
    if (in_array($this->environment, array('dev', 'test'))) {
        return '/dev/shm/litwicki/cache/' . $this->environment;
    }

    return parent::getCacheDir();
}
```

```

public function getLogDir()
{
    if (in_array($this->environment, array('dev', 'test'))) {
        return '/dev/shm/litwicki/logs';
    }

    return parent::getLogDir();
}

```

Write a CacheCommand you can love and cherish forever and ever

```

<?php namespace LitwickiAppBundleCommand;

use SymfonyComponentConsoleInputInputArgument;
use SymfonyComponentConsoleInputInputInterface;
use SymfonyComponentConsoleOutputOutputInterface;
use SymfonyComponentConsoleInputInputOption;

use SymfonyComponentFilesystemFilesystem;
use SymfonyComponentProcessProcess;
use SymfonyComponentProcessExceptionProcessFailedException;

use SymfonyBundleFrameworkBundleCommandContainerAwareCommand;

class ClearCacheCommand extends ContainerAwareCommand
{
    protected $interval = 5;
    /**
     * @var array
     */
    protected $config = array();
}

```

```

/**
 * @var array
 */
protected $connections = array();

protected function configure()
{
    $this->setName('cache:clear')
        ->setDescription('Clear the application cache and reset
permissions.')
    ;
}

public function execute(InputInterface $input, OutputInterface
$output)
{

    $root = '/dev/shm/litwicki';
    $cache = sprintf('%s/cache', $root);
    $logs = sprintf('%s/logs', $root);

    /**
     * Remove the entire cache/logs directory
     */
    $process = new Process(sprintf('sudo rm -rf %s', $root));
    $process->run();

    // executes after the command finishes
    if (!$process->isSuccessful()) {
        throw new ProcessFailedException($process);
    }
}

```

```

/**
 * Recreate the cache/logs home dir
 */
$process = new Process(sprintf('sudo mkdir %s', $root));
$process->run();

// executes after the command finishes
if (!$process->isSuccessful()) {
    throw new ProcessFailedException($process);
}

/**
 * Reset permissions on cache/logs home
 */
$process = new Process(sprintf('sudo chmod -R 0777 %s',
$root));
$process->run();

// executes after the command finishes
if (!$process->isSuccessful()) {
    throw new ProcessFailedException($process);
}

$output->writeln('');
$output->writeln('<info> LITWICKI: cache / logs rebuild
complete</info>');
$output->writeln('');

}

protected function waitInterval()

```

```
{  
    sleep($this->interval);  
}  
}
```

PARTY!!!

Now you can enjoy life and all references to `cache:clear` will run your new and improved command that will also work noticeably faster in Vagrant development environments; thanks *entirely* to [Benjamin Eberlei](#) and his wonderful [blog post](#).

Enjoy!

```
cd /path/to/myapp && php bin/console cache:clear
```