Write a console command

A task guide for a command line tool in an extension - the command class extending Exponential\Runnable\Command, the entry file, options and help, siteaccess and database, output and exit codes, the @description tag...

Source file: doc/eng-us/help/php-developers/write-a-console-command.md Last changed 2026-11-04
https://doc.exponential.earth/exphelp/viewer/help/php-developers/write-a-console-command.md

This task guide is for PHP developers who add a command line tool to an extension: an import, a report, a repair script. The command is a class extending `Exponential\Runnable\Command`; a short entry file calls it; the console `bin/php/console` finds it without any registration.

Contents

1. The result
2. Step 1: the command class
3. Step 2: the entry file
4. Step 3: autoloads and a first run
5. Options and arguments
6. The helpers
7. Output, logging and exit codes
8. Running as root, siteaccesses and users
9. Long-running commands

The result

Shell

```
php bin/php/console ext:acmereport:report --month=2026-09
php bin/php/console list ext:acmereport            # every command of the extension
php bin/php/console ext:acmereport:report --help
```

Step 1: the command class

`extension/acmereport/classes/runnable/commands/report.php`:

PHP

```
<?php
namespace Exponential\Command\Extension\Acmereport;

class Report extends \Exponential\Runnable\Command
{
    public function run()
    {
        $this->script( array( 'description' => "Writes the monthly report of acmereport",
                              'use-session' => false, 'use-modules' => true, 'use-extensions' =>
true ) );
        $options = $this->startup( '[month:]', '', array( 'month' => 'The month, YYYY-MM' ) );
```

```
        $month = (string)$options['month'];
        if ( !preg_match( '/^\d{4}-\d{2}$/', $month ) )
        {
            $this->error( 'Give the month as --month=YYYY-MM' );
            $this->shutdown( 1 );
        }

        $totals = \acmeReportService::monthlyTotals( $month );   // the work: a class you can cal
l anywhere
        foreach ( $totals as $status => $total )
            $this->output( sprintf( '%-12s %8d', $status, $total ) );

        $this->shutdown( 0 );
    }
}
```

Step 2: the entry file

`extension/acmereport/bin/php/report.php`:

PHP

```php
#!/usr/bin/env php
<?php
/**
 * @description Writes the monthly report of acmereport
 */
require_once 'autoload.php';
\Exponential\Command\Extension\Acmereport\Report::main( __FILE__ );
```

The `@description` tag is what `php bin/php/console list` shows; a `@long-description` tag adds text to the help. Make the file executable (`chmod +x`).

Step 3: autoloads and a first run

Shell

```shell
php -l extension/acmereport/classes/runnable/commands/report.php
php bin/php/ezpgenerateautoloads.php -e
php bin/php/console ext:acmereport:report --help
php bin/php/console ext:acmereport:report --month=2026-09
```

Run every command from the installation root (the directory with `index.php`); `require_once 'autoload.php'` relies on it.

Options and arguments

The option string is the format of `eZCLI::parseOptionString()`, which `eZScript::getOptions()` uses:

Form	Meaning
`[o]`	a short option, `-o`
`[opt]`	a long option, `--opt`
`[o|opt]`	the same option in short or long form

Form	Meaning
`[opt:]`	an option with a value, `--opt=value`
`[opt;]`	an optional value, `--opt` or `--opt=value`
`[opt*]`, `[opt+]`, `[opt?]`	given any number of times, at least once, at most once

Read the values from the returned array by the long name (`$options['month']`), and check an option that may be missing with `empty()` before you use it.

The second argument describes positional arguments; the third gives the help text of each option. The standard options come for free: `--help`, `-q`/`--quiet`, `-s`/`--siteaccess`, `-l`/`--login`, `-d`/`--debug`, `-v`/`--verbose`.

The helpers

Helper	Does
`$this->script( $settings )`	creates the script, `eZScript::instance( $settings )`
`$this->startup( $config, $argumentConfig, $optionHelp )`	the script's `startup()`, the options, the script's `initialize()` (siteaccess, extensions, session, database)
`$this->options( ... )`	only the option parsing, when you need steps in between
`$this->cli()`	the terminal, `eZCLI::instance()`
`$this->output()`, `error()`, `warning()`	a line, an error, a warning; `output()` writes nothing with `-q`
`$this->isQuiet()`	`-q` was given
`$this->log( $message, $logName = null )`	`eZLog::write()`, to `var/log/<script name>.log` by default
`$this->shutdown( $exitCode, $exitText )`	`eZScript::shutdown()`, which exits

Output, logging and exit codes

- Exit with `0` on success and another code on failure: deploy scripts, cron and CI read it, and the console passes it on unchanged.
- Write results to standard output and problems through `error()`, so a pipe receives only the results.
- Log what changed data (`$this->log()`), never passwords, tokens or personal data.
- Offer `--dry-run` for anything that changes data.

Running as root, siteaccesses and users

- Scripts refuse to run as `root` unless given `--allow-root-user`. Run them as the site's user where you can, so files they write in `var/` belong to the site.
- `-s <siteaccess>` picks the settings; without it the default siteaccess is used.
- In a script the current user is anonymous until the script logs one in. Use a dedicated user with exactly the policies the command needs, not the administrator.

Long-running commands

- Fetch content in pages and clear in-memory caches between pages (`eZContentObject::clearCache()`), or memory grows with every object.
- Commit in batches with short transactions.
- Print progress unless `-q`; cron runs with `-q`.
- Commands that should run on a schedule are better as a cronjob part that calls the same service (Write a cronjob part).

Related pages

- Module views as classes
- Exponential Console
- Console commands for developers
- Free documentation: this page is free to copy, translate and improve under the GNU FDL 1.3.

> **Become an Exponential Partner**
>
> Partners share their information, provide funding, contribute back and develop the products with us, and that is how Exponential grows. A partner website and dashboard are planned for mid-2027: please become a partner today to speed up our growth rate and land new features faster.
>
> Partners: start here | How to apply

Exponential Copyright © 1998-2026 Exponential Foundation and 7x
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".
This documentation is free documentation, licensed under the GNU Free Documentation License 1.3 or later:
https://www.gnu.org/licenses/fdl-1.3.html.