
Daedalus Documentation

Release 0.1.0

Joshua Estes

Sep 27, 2017

Contents

1	Installation	3
1.1	With Composer	3
1.2	Global Install with Composer	3
1.3	Without Composer	3
2	Buildfile Format	5
2.1	Basic Example	5
2.2	Advanced Example	6
3	Properties File Format	7
3.1	Default Properties	7
3.2	Usage	7
3.3	Comments	7
3.4	Environment Variables	7
3.5	Example Properties File	8
4	Usage	9
4.1	Getting Help	9
5	Tasks	11
5.1	Defining Tasks	11
6	Commands	13
6.1	chgrp	13
6.2	chmod	13
6.3	chown	14
6.4	copy	14
6.5	exec	15
6.6	mirror	15
6.7	mkdir	15
6.8	phar	15
6.9	phpcs	15
6.10	phplint	15
6.11	phpunit	16
6.12	remove	16
6.13	rename	16
6.14	symlink	16

6.15	touch	16
6.16	HOWTO Create a New Command	16
7	Configuration	17
8	Style Guide	19
8.1	Output	19
8.2	Colors	19
8.3	Return Codes	19
9	Changelog	21
9.1	1.0.0	21
10	License	23
11	Helpful Linkage	25

Daedalus is a PHP Build Tool that aims to be easy to setup and configure builds for your projects.

CHAPTER 1

Installation

Daedalus can easily be installed using [Composer](#).

With Composer

```
php composer.phar require "daedalus/daedalus ~1.0@dev"
```

Global Install with Composer

```
php composer.phar global require "daedalus/daedalus ~1.0@dev"
```

Note: You need to make sure that the global composer bin directory is in your PATH

Without Composer

It's not recommended that you work with Daedalus without the use of Composer since the project takes advantage of Composer autoloading functionality. If however you want to install Daedalus into a project without Composer you will need to make sure you have the required dependencies which you can find in the `composer.json` file and you will need to make sure you have an autoloader setup. All of which is beyond the scope of this project.

CHAPTER 2

Buildfile Format

Daedalus uses the concept of a Task and a Task runs multiple commands. See the Commands section for an overview of the commands that you can use.

Basic Example

```
# build.yml
daedalus:
  tasks:
    <taskname>:
      description: <description>
      commands:
        <command name>:
          command: <command name>
          arguments:
            <argument>: <argument value>
          options:
            <option>: <option value>
```

Note: Arguments are based on the command that you choose to run. For a list of arguments for the command, please refer to the Commands section.

The `<taskname>` can be replaced with any unique name that you choose. This will also be the task that you run from the command line ie `bin/daedalus <taskname>`.

The `<description>` is optional, but if you add one, it will be displayed when the tasks are listed.

You can have as many commands as you want and they will be ran in the order in which they are added.

Complete Basic Example

```
# build.yml
daedalus:
  tasks:
    chmod:
      description: Run chmod on a file
      commands:
        chmod:
          command: chmod
          arguments:
            mode: 0755
            file: /path/to/file
```

Advanced Example

You can configure various settings of Daedalus to make the behaviour different. Please refer to the Configuration section for a complete listing of these settings.

Complete Advanced Example

Properties File Format

You use properties files in case you want to have a different build based on where you are running the tool at. For example, you want to have a few different parameters when running a build on Travis CI than you would your local development machine.

Default Properties

Property	Value
user.home	\$HOME

Usage

In your build file, put the property between a percentage sign. For example `%user.home%`.

Comments

Any line that starts with `#` is considered a comment.

Environment Variables

TODO

Example Properties File

```
# All properties files must have the extension .properties or they
# will not load.
# Empty lines and lines starting with "#" are not processed

property_name=property

# You can also use other properties
build_dir=%user.home%/build
```

CHAPTER 4

Usage

Once installed and your build file setup, you can simply run daedalus to get a list of tasks you can run.

```
vendor/bin/daedalus
```

Getting Help

If you ever need help for any of the commands, you can run the help command which will return the help document on how to use that command. For example

```
vendor/bin/daedalus help chmod
```

You can also get an idea of the properties you have access to and some other information by running the dump-container task.

```
$ vendor/bin/daedalus dump-container
+-----+-----+
| Parameter Name | Parameter Value |
+-----+-----+
| env.editor     | vim             |
| env.home       | /Users/joshuaestes |
| env.lang       | en_US.UTF-8     |
| env.pwd        | /Users/joshuaestes/Projects/joshuaestes/Daedalus |
| env.shell      | /bin/bash       |
| env.tmpdir     | /var/folders/w_/vb3pk8dj5tq47b37n6nmzxbh0000gn/T/ |
| env.user       | joshuaestes     |
| php.version    | 5.4.33          |
+-----+-----+

+-----+-----+
| Service ID     | Class           |
+-----+-----+
| application    | Daedalus\Application |
+-----+-----+
```

command.chgrp	Daedalus\Command\ChgrpCommand	
command.chmod	Daedalus\Command\ChmodCommand	
command.chown	Daedalus\Command\ChownCommand	
command.copy	Daedalus\Command\CopyCommand	
command.dump_file	Daedalus\Command\DumpFileCommand	
command.exec	Daedalus\Command\ExecCommand	
command.help	Daedalus\Command\HelpCommand	
command.mirror	Daedalus\Command\MirrorCommand	
command.mkdir	Daedalus\Command\MkdirCommand	
command.phar	Daedalus\Command\PharCommand	
command.phplint	Daedalus\Command\PhpLintCommand	
command.remove	Daedalus\Command\RemoveCommand	
command.rename	Daedalus\Command\RenameCommand	
command.symlink	Daedalus\Command\SymlinkCommand	
command.touch	Daedalus\Command\TouchCommand	
event_dispatcher	Symfony\Component\EventDispatcher\EventDispatcher	
filesystem	Symfony\Component\Filesystem\Filesystem	
finder	Symfony\Component\Finder\Finder	
kernel	Daedalus\Kernel	
process_builder	Symfony\Component\Process\ProcessBuilder	
service_container	Symfony\Component\DependencyInjection\ContainerBuilder	
task.build	Symfony\Component\Console\Command\Command	
task.checkstyle	Symfony\Component\Console\Command\Command	
task.lint	Symfony\Component\Console\Command\Command	
task.phpunit	Symfony\Component\Console\Command\Command	
+-----+-----+-----+		

As you can see there is a section of properties and a section of services. The services are helpful if you ever want to develop or hack Daedalus.

Tasks are just a simple list of various commands that are ran.

Defining Tasks

```
<taskname>:
  description: <description>
  commands:
    <command name>:
      command: <command name>
      arguments:
        <argument name>: <argument value>
```


Commands are ran inside of a task.

chgrp

chmod

Allows you to chmod a file or directory.

Arguments

file

This is either the full path to a file or a directory.

mode

Mode which you want to set.

Warning: This MUST be in octal format, 755 and 0755 will yeild different results.

Example Usage

```
daedalus:
  tasks:
    chmod:
```

```
commands:
  chmod_da_file:
    command: chmod
    arguments:
      file: %user.home%/.daedalus/
      mode: 0744
```

chown

copy

Copy file from one location to another

Arguments

src

Source location of file

dest

Destination of where to put file

Options

overwrite

If the file already exists, replace it.

Examples

```
daedalus:
  task:
    copy_file:
      commands:
        first_command:
          command: copy
          arguments:
            src: /path/to/file.ext
            dest: /path/to/copied/file.ext
          options:
            overwrite: true
```

exec

mirror

mkdir

phar

phpcs

phplint

Runs lint checking on PHP Files.

Arguments

source

This can be a list of files or directories.

Examples

Basic example that shows how to use the basic functionality.

```
daedalus:
  tasks:
    lint_ya_files:
      commands:
        linting_example:
          command: phplint
          arguments:
            source: 'src/'
```

In this example it shows that you can use an array of sources.

```
daedalus:
  tasks:
    lint_ya_files:
      commands:
        linting_example:
          command: phplint
          arguments:
            source: ['src/', 'lib/']
```

phpunit

remove

rename

symlink

touch

HOWTO Create a New Command

CHAPTER 7

Configuration

Various configuration settings that are used by the tool itself.

This guide **SHOULD** provide a guide on how to display output for commands that run for a build and what type of return codes **SHOULD** be returned for success or failure.

Output

Colors

Return Codes

Return Code	Status
0	Successful
-1	Failure

CHAPTER 9

Changelog

1.0.0

- First stable release

CHAPTER 10

License

The MIT License (MIT)

Copyright (c) 2014 Joshua Estes

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CHAPTER 11

Helpful Linkage

- [Homepage](#)
- [Issues](#)
- [Travis CI](#)
- [Sensio Labs Insight](#)