

# Manage WordPress with Awesome using wp-cli

WordCamp Vegas 2012

Mike Schroder (DH-Shredder)  
@GetSource - <http://www.getsource.net>

# Who Am I?

- Mike Schroder, a.k.a DH-Shredder, a.k.a. @GetSource
- Third Culture Kid, enjoy Coffee & Sailing
- WordPress Core and wp-cli Contributor
- Happy DreamHost Employee

***There are two groups  
of people.***

***Those who use the  
command line***

***Those who are going to use  
the command line***

***Don't be afraid of the CLI.***

**It's your friend.**

*Oh, you like the CLI?*  
wp-cli will make your life better.

# *What's wp-cli?*

super-cool *Open Source* tool  
to manage WordPress



# *Why so cool?*

Headed up by *Andreas Creten* and  
*Cristi Burcă (scribu)*

# *Why so cool?*

Uses *WordPress itself* to perform operations

*Why so cool?*

Automation!

***What can I do with it?***



***No, Really.***

# *Update WordPress*

**wp core update**

# *Install a Theme*

**wp theme install sunspot**



# *Reset to default theme*

**wp theme activate twentytwelve**

# *Backup your Database*

```
wp db export backup.sql
```

# *Update Plugins*

```
wp plugin update --all
```

# What do I need to run it?

# What do I need to run it?

- SSH access to your WordPress install's directory

# What do I need to run it?

- SSH access to your WordPress install's directory
- PHP 5.3+

# What do I need to run it?

- SSH access to your WordPress install's directory
- PHP 5.3+
- WordPress 3.3+

# What do I need to run it?

- SSH access to your WordPress install's directory
- PHP 5.3+
- WordPress 3.3+
- Enough RAM for shell processes to run WordPress



# What do I need to run it?

- SSH access to your WordPress install's directory
- PHP 5.3+
- WordPress 3.3+
- Enough RAM for shell processes to run WordPress
- Easiest on Linux & MacOS

*Okay.* Got that covered.  
How can I get this  
***Awesomeness?***

# *Download wp-cli*

```
git clone --recursive git://github.com/wp-cli/wp-cli.git
```

***Make it runnable from your  
WordPress Install.***

*If you have sudo:*

**sudo utils/dev-build**

*Otherwise, add an alias  
(.bashrc/.bash\_profile)*

```
alias wp=' /home/user/wp-cli/src/bin/wp ' ;
```

*In ~/.bash\_profile:*

```
if [ -f ~/.bashrc ]; then  
    source ~/.bashrc  
fi
```

([http://www.joshstaiger.org/archives/2005/07/bash\\_profile\\_vs.html](http://www.joshstaiger.org/archives/2005/07/bash_profile_vs.html))

***You've got it installed?***

**Let's dig deeper.**



***wp-cli is extensible.***

*Sample Plugin:*

WCLV Backup.

# *Goal:*

*wp wclv backup [--no-db] [/dir/outputfile.tar.gz]*

## *Our Plan:*

- Use built-in SQL Backup
- Create a .tar.gz of install and db

# Define the Base Command

```
<?php

// Let WP_CLI know we exist!
// Earlier versions of wp-cli used WP_CLI::addCommand()
WP_CLI::add_command( 'wclv', 'WCLV_Backup_Command' );

/**
 * The WCLV Backup Plugin
 *
 * @package WCLV_Backup
 * @subpackage commands/community
 * @maintainer Mike Schroder
 */
class WCLV_Backup_Command extends WP_CLI_Command {
    . . .
}
```

# Define Sub-Commands

- `$args`: stand-alone arguments
- `$assoc_args`: `--arg=value` style in associative array

```
class WCLV_Backup_Command extends WP_CLI_Command {  
  
    function backup( $args, $assoc_args ) {  
        $filename = $dbname = null;  
        ...  
    }  
  
    public static function help() {  
        WP_CLI::line( "usage: wp wclv backup [--no-db] [path/to/file]" );  
    }  
}
```

# Grab Filename

```
function backup( $args, $assoc_args ) {
    $filename = $dbname = null;

    // If a filename isn't specified, default to "Site's Title.tar.gz".
    if ( empty( $args ) )
        $filename = '../' . escapeshellarg( get_bloginfo() ) . '.tar.gz';
    else
        $filename = $args[0];
}
```

# Handle --no-db

- SQL file not using temp location for simplicity of demo.

```
// If --no-db is specified, don't include the database in backup
if ( ! isset( $assoc_args['no-db'] ) ) {
    $dbname = '../database_temp.sql';

    // This is cheating a bit, since wp-cli doesn't currently support
    // running commands within commands without re-launching itself.
    WP_CLI::run_command( array( 'db', 'export', $dbname ), array() );
}
```



# Back it up!

- See class-wp-cli.php for more magical functions

```
// GZ/Tar and Backup the install!
WP_CLI::line( "Backing up to '$filename' ..." );
$result = WP_CLI::launch( "tar -zcvf $filename . $dbname", false );

// If we created a database backup, remove the temp file.
if ( $dbname && ! unlink( $dbname ) )
    WP_CLI::warning( "Couldn't remove temporary database backup,
'$dbname'." );
```

# ERROR ERROR

```
// Will automatically exit on WP_CLI::error, but not WP_CLI::success.  
if ( 0 == $result ) {  
    WP_CLI::success( "Backup Complete." );  
} else {  
    WP_CLI::error( "Backup Failed." );  
}  
}
```

# Resources!

- <https://github.com/wp-cli/wp-cli>
- <https://github.com/wp-cli/wp-cli/wiki/List-of-internal-commands>
- <https://github.com/wp-cli/wp-cli/wiki/Commands-Cookbook>
- <http://scribu.net/wordpress/a-command-line-interface-for-wordpress.html>
- <http://wp.tutsplus.com/tutorials/using-wp-cli-for-fun-and-profit/>
- <http://halfelf.org/2012/command-line-wp/>
- [http://www.joshstaiger.org/archives/2005/07/bash\\_profile\\_vs.html](http://www.joshstaiger.org/archives/2005/07/bash_profile_vs.html)