



- - -

Version 1.0

- - -

Documentation

Frédéric BOLLON

October 10, 2009

Contents

1	Introduction	2
2	Requirements	3
3	Installation	4
4	Usage	6
4.1	Setting up your project	6
4.2	Deployment steps	9
4.2.1	Adding a new project	9
4.2.2	Project deployment	10
4.2.3	Project deployment fast mode	11
4.3	Deployment logs	11
4.4	Logs viewer	11
4.5	Tips and tricks	11
4.5.1	Common to all kinds of project	11
4.5.2	Specific to CakePHP projects	13
5	Related links	14

Chapter 1

Introduction

Fredistrano is a deployment tool for PHP web applications. It automates the export of your sources from a subversion repository and synchronizes them with the content of a target directory. Several manual tasks are also handled directly by the application during the deployment: renaming of configuration files, modifying permissions... Fredistrano is inspired from the smart Capistrano project.

From a technical perspective, this application has been written in PHP with the cakePHP framework. Fredistrano both supports Windows/Linux environment. Concerning Windows, Fredistrano requires the installation of Cygwin in order to use the rsync command.

Special thanks to Aurélien Millet and euphrate_ylb for their help in this project.

Chapter 2

Requirements

Fredistrano must be installed on a Apache Web server hosted **indifferently** under Linux or Windows.

Common to all OS

- A Php project "versioned" with Subversion
- A Php web hosting with the `safe_mode` set to Off and `mod_rewrite` activated

If mod_rewrite is not working you will need to remove the comment from `app/config/core.php`

a. Around line 59: `Configure::write('App.baseUrl', env('SCRIPT_NAME'));`

b. Then access using `http://www.example.com/my_directory/index.php/pages` to verify installation is working

Windows only

- For deployment on windows servers, you need to install cygwin with packages `rsync`, `perl` and `subversion`.

Chapter 3

Installation

- Download the latest version of Fredistrano on <http://code.google.com/p/fredistrano/downloads/list>
- Untar the archive at the root of your web server or in a directory of your choice
`tar xzvf fredistrano_x.x.tar.gz`
- Give writable permission to temporaries directories (only on linux servers).

```
chmod -R 777 app/tmp/ files/
```

- Create the database with the sql script in `/app/config/sql/fredistrano.sql`
- Create files `app/config/config.php` et `app/config/database.php` with the help of `config.prd.php` and `database.prd.php`

- Configure app/config/database.php following your database

example :

```
var $default ← array(  
    'driver' ↔ 'mysql',  
    'persistent' ↔ false,  
    'host' ↔ 'localhost',  
    'login' ↔ 'usermysql',  
    'password' ↔ 'password',  
    'database' ↔ 'fredistrano',  
    'encoding' ↔ 'utf8'  
);
```

- A user "admin" with password "pass" is created by the sql script, don't forget to change password in Administration/users. If you create a new user, you need to affect if the group "admin". **Do not delete the user "admin" before creating a new user for the group "admin".**

Chapter 4

Usage

4.1 Setting up your project

Before being able to deploy your project with fredistrano, you must set it up according to the following recommendations :

- Your project must be versionned with subversion.
- If the content of some files differs between the deveploment environment and the production one, these files shouldn't be added to subversion but only copies with an extension like '.dev.xxx' et '.prd.xxx':
For example, if the application requires a database.php file where connection parameters are defined, you won't add directly this file to subversion. Instead, you will add two other versions, one for each environment, nammed database.dev.php and database.prd.php.
During the deployment, whatever the extension is, all 'prd.xxx' files will be deployed (not 'dev.xxx' ones) and renamed in '.xxx'.
- A directory ".fredistrano" must be created at the application root. In this directory you must create a file "deploy.php". This files contains application specific deployment parameters such as a list of files and folders that shouldn't be synchronized, a list of writable directories... Notice that this file won't be deployed in your production environment. Here is an example of deploy.php :

```
<?php
class DEPLOY_CONFIG {
```

```

// Default deployment options for the current
// project
// these options could be modified during the
// deployment process in standard mode
// in fast mode these options will be used
var $options = array(
    'export'      => array(),
    'synchronize' => array(
        'runBeforeScript' => false,    //enable
        custom script before deployment
        'backup'      => false    //enable
        backup functionality
    ),
    'finalize'    => array(
        'renamePrdFile'      => false,    //enable
        renaming .prd.xxx files
        'changeFileMode'    => false,    //enable
        updating file mode
        'giveWriteMode'     => false,    //enable
        updating write mode on directories
        defined in $writable (in this file)
        'runAfterScript'    => false    //enable
        custom script at the end of the
        deployment process
    )
);

// path of yours custom scripts to execute at
// the beginning/end of the deployment process
// if yours scripts are located in a directory
// named ".fredistrano" at the root of your
// project enter only the name of your script
// to execute
var $scripts = array(
    'before' => '/path/to/file',
    'after'  => '/path/to/file'
);

// List of directories and files to exclude
// during the deployment process on the

```



```
        production server
var $exclude = array (
    '/app/tmp/logs/*',
    '/app/tmp/sessions/*',
    '/app/tmp/tests/*',
    '/app/config/database.php',
    '/app/config/config.php',
    '/app/webroot/files/*',
    '/files/logs/*',
    '/files/tmp/*'
);

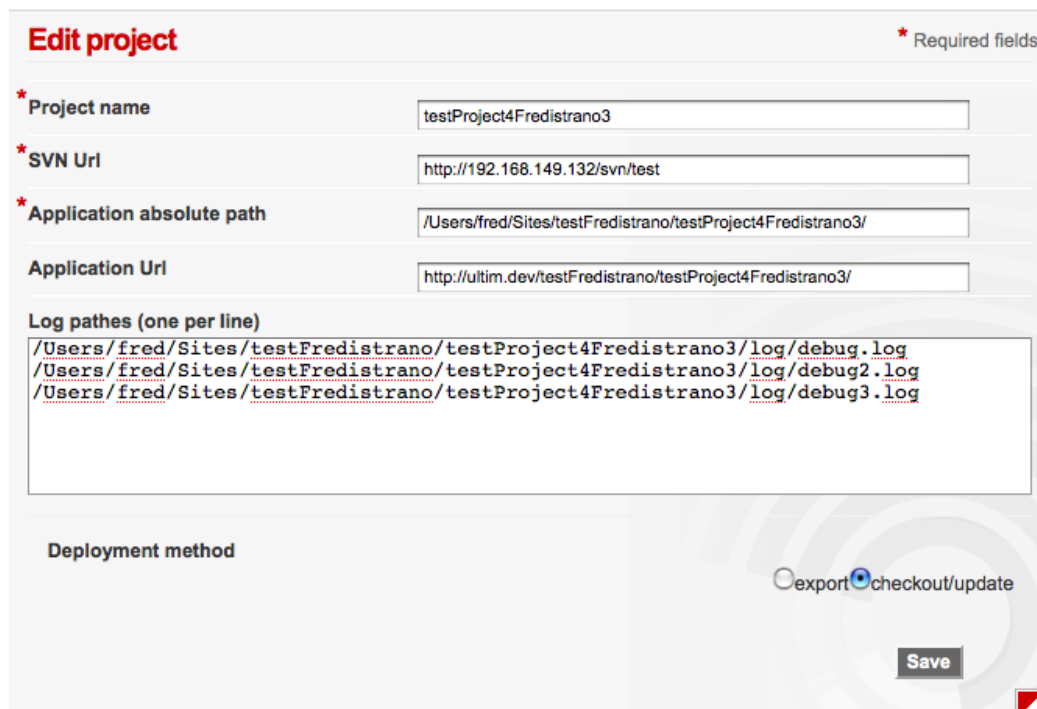
// Directories list on which the write
// permission will applied during the
// finalization step of the deployment process
// log, cache, upload directories, etc...
var $writable = array (
    '/app/tmp',
    '/files/backup',
    '/files/logs'
);

} // DEPLOY_CONFIG
?>
```

4.2 Deployment steps

4.2.1 Adding a new project

To deploy a PHP web application with Fredistrano, the first step is to create a new project.



Edit project * Required fields

* **Project name**

* **SVN Url**

* **Application absolute path**

Application Url

Log paths (one per line)

```
/Users/fred/Sites/testFredistrano/testProject4Fredistrano3/log/debug.log
/Users/fred/Sites/testFredistrano/testProject4Fredistrano3/log/debug2.log
/Users/fred/Sites/testFredistrano/testProject4Fredistrano3/log/debug3.log
```

Deployment method

☐ export ☒ checkout/update

Save

1. Click on the "Projects" toggle (Top of your screen)
2. Click on the "Add a new project" link
3. Fill out the form as follow:
 - Field: "**Project name**" : Is used to identify the project. It will also be used as a directory name for temporary copy during the deployment. Do not use special characters.
 - Field: "**SVN Url**" : Url of the SVN repository of the project that should be deployed (eg. : "http://svn.mondomain.com/monProjet/trunk").
 - Field: "**Application Url**" : Won't be used by Fredistrano. Juste a simple reminder.

- Field: **"Absolute path"** : Absolute path of the target folder where the application should be deployed
 Example for a Windows server (located on the server) : D:\www\html\myProject
 Example on a Linux server : /var/www/html/monProjet or /home-/monUser/monDomain.com.
- Field: **"Log pathes"** : Pathes of files that you want to access with the log viewer, one path per line.
- Field: **"Deployment method"** : **Checkout/update**: fastest method, during the first deployment the command "svn checkout" will be executed for the next deployment the command "svn update" will be used to retrieve only modified files. **Export**: slower, because for each deployment the command "svn export" will be used, however it is sometimes necessary to use this method, for example if you do not have the necessary privileges on the subversion repository to execute the checkout command.

4.2.2 Project deployment

1. Display the details of the project that should be deployed.
2. Click on "Deploy project".
3. Specify the revision number. If empty, Fredistrano will checkout the latest revision of the repository.
4. In the case of a private Subversion project protected by login/password, you may specify your credentials at this step. If all your projects are using the same credentials, you may specify them in the file *app/config/config.php*.
5. Click on "Step 1 SVN export". An svn export is executed and the result is displayed in the console located at the bottom (*output console*) "Fredistrano/files/tmp/nomduprojet/tempDir".
6. For the next step, if "simulate" is checked, the rsync command will only be simulated and its result may be seen in the output console.
7. Click on "Step 2 synchronize".
8. A backup is performed in "Fredistrano/files/backup/projectName". The rsync command is executed between "Fredistrano/files/tmp/projectName/tempDir" and "projectName".

9. Finally, as the last step, three options may be overridden: 1. renaming of '.prd.' files, 2. force files and folders permissions, 3. set write privileges on special folders . The initial values depend on the content of your config.php file.
10. Click on "Step 3 finalize" and you're done!

4.2.3 Project deployment fast mode

1. Display the details of the project that should be deployed.
2. Click on the "switch deployment mode" link to change the button "Deploy" to "Fast deploy"
3. All the steps will be executed with standard options (renaming prd files, changing file mode on modified files only, giving write mode).

4.3 Deployment logs

You can access the deployment history from the details page of each project by clicking on the link "view deployment history". A rss feed of deployment history is available in the menu on this page. It is possible to disable rss feeds in /app/config/config.php

4.4 Logs viewer

Click on the "Logs" tab to use the log viewer, with this feature you can display the content of the logs files defined in the project details

4.5 Tips and tricks

4.5.1 Common to all kinds of project

- **Script example for "beforeScript"**

This script makes it possible to replace the .htaccess of a project by another .htaccess during the deployment synchronisation step which will redirect the visitors towards a temporary page. The temporary .htaccess file and the page which will be used for the redirection are placed in the .freistrano directory at the project root.

```
#!/bin/sh

PATHTMP=/Path/on/production/server/fredistrano/
files/tmp/myproject/tmpDir
PATHPRD=/Path/on/production/server/myproject

# copy the specific temporary .htaccess at the
right place
cp -vf ${PATHTMP}/.fredistrano/.htaccess ${PATHPRD}
}/.htaccess
# copy the specific temporary redirect page at
the right place
cp -vf ${PATHTMP}/.fredistrano/302.html ${PATHPRD}
}/302.html
```

- **Script example for "afterScript"**

This script will change a value in a file, in this example it will modify the value of the debug mode to 0 in the file `app/config/core.php`. It also will restore the `.htaccess` file which one replaced by a temporary file during the deployment and will remove the redirection file (see the example of script `beforeScript`)

```
#!/bin/sh
# modify the debug mode from 1 or 2 to 0 in file
app/config/core.php
PATHTMP=/Path/on/production/server/fredistrano/
files/tmp/myproject/tmpDir
PATHPRD=/Path/on/production/server/myproject

sed -i .bak "s/\('debug',\) [ ]*[12]/\1_0/g" ${
PATHPRD}/app/config/core.php

# replace the temporary .htaccess
cp -vf ${PATHTMP}/.htaccess ${PATHPRD}/.htaccess

# remove the temporary redirection page
rm -vf ${PATHPRD}/302.html
# remove the old version of app/config/core.php.bak
rm -vf ${PATHPRD}/app/config/core.php.bak
```

4.5.2 Specific to CakePHP projects

- **Removal cache:** To empty the cache automatically during each deployment, you have to version the directories `app/tmp/cache/models`, `app/tmp/cache/persistent` and `app/tmp/cache/views` and ignore their content. In this way, during the deployment synchronization stage these directories will be emptied on the production server.

To ignore the content of these directories I use the following command at the project root in the working copy :

```
svn propset svn:ignore "*" app/tmp/cache/models/  
app/tmp/cache/persistent/ app/tmp/cache/views/  
app/tmp/logs/ app/tmp/sessions/ app/tmp/tests/
```

Chapter 5

Related links

You will find in the following

- An example of web host that satisfy the requirements of Fredistrano
<http://www.fbollon.net/node/12>
- A quick explanation on how to use Subversion
<http://www.fbollon.net/node/65>
- The official support forum of Fredistrano
<http://groups.google.com/group/fredistrano-discuss>